Design and Implementation of a Circuit for Mesh Networks with Application in Body Area Networks

Fardin Derogarian
MAP-Tele, INESC TEC
Faculdade de Engenharia, Universidade do Porto
Porto, Portugal
Email: mpt09020@fe.up.pt

João Canas Ferreira and Vítor M. Grade Tavares
INESC TEC
Faculdade de Engenharia, Universidade do Porto
Porto, Portugal
Email: jcf@fe.up.pt and vgt@fe.up.pt

***Abstract*—This paper presents a network circuit for wearable low-power BAN (Body Area Networks) applications, geared towards mesh network topologies with conductive yarns as transmission channels. The design and implementation of the physical and MAC layers is described. The resulting circuit sends and receives data simultaneously, and experimental results indicate that the proposed system works with variable data rates, up to a maximum of 9+9 Mbps. All reported measurements were collected from working FPGA-based prototypes, and the performance achieved shows that the circuit is suitable for use in reliable high-speed low-power BAN applications.**

I. Introduction

An important aspect of the implementation of a Wearable Health Systems (WHS), as a specific category of Personal Health Systems (PHS), is the underlying body area network (BAN) or wearable network [1]–[3]. In general, BANs can be wireless or wired networks [4]. Based on the application requirements, the frequency band used by the wireless network can be selected from a diverse set of applicable categories and technologies such as ZigBee, Bluetooth, and Wireless LAN (WLAN). These approaches enjoy the benefits of wireless networks, but may suffer from interference, fading, and low data-rates [4], [5]. Reliable communication is an important parameter in BAN applications [6]. Usually wireless BANs operate in the industrial, scientific and medical (ISM) band, which exposes them to increased interference from other systems that work in the same band. In wireless systems, like Bluetooth, communication is performed over pre-established links. Any link failure entails a complex recovery procedure, which takes more power and decreases reliability.

Wired networks, as a second type of communication infrastructure for BAN applications, are available to provide high speed, reliable and low-power solutions [7]–[9]. The use of fabrics with embedded conductors can provide the required electrical connections, and would make the product more user-friendly and comfortable. Previous developments and on-going work show that the most comfortable and easiest way to monitor physiologic signals consists in using garments [2], [3], with conductive yarns used as wire lines. The design of such a wearable system must take into account the fact that electrical characteristics of conductive yarns are not exactly those of normal wires, since their electrical properties may exhibit significant variations in relaxed and stretched modes. In addition, conductive yarns are prone to tearing and wearing out; thus, to achieve high reliability the network must be tolerant to catastrophic and parametric faults.

A typical BAN system has a central module that collects data from the sensors. The connections of sensor nodes to the central module can have different topologies, such as bus, star or mesh. In a bus connection, all sensors are connected to a single shared bus, and the central node controls all communication. Each additional sensor increases the load on the connection bus, resulting in increased power consumption and reduced data rate. The lack of alternative paths and the vulnerability of conductive yarns decrease the reliability of the this topology. In star arrangement, each sensor directly connects to the central node. This is not a scalable solution, because the central module must have enough ports to connect to every sensor, a situation that becomes more impractical as the number of sensors increases. In contrast, a mesh network typically has several alternative interconnections between sensors; if any link or sensor fails, there are many other ways for two healthy nodes to communicate. The mesh topology is the one offering the best advantages especially for large numbers of network nodes.

This paper describes the design and implementation of the physical (PHY) and media access control (MAC) layers for a mesh-topology BAN. The module will be used to implement a sensor network that captures electromyographic signal and movement information from the lower limbs of walking subjects. The collected information will be used for the analysis of normal or pathological gait in order to support medical diagnosis, and for pre- and post-assessing treatment programs. The rest of the paper is organized as follows: Section II describes the related works. Section III describes the performance of the system and section IV explains the prototype architecture. Section V presents and discusses the results obtained with the prototype. Section VI sums up the main conclusions drawn from this work.

II. Related Work

A number of wired networks for wearable applications have been proposed. Using the conductive yarns as a data bus in textiles was introduced by Post and Orth [10]. Wade and

Asada [11] introduced a wearable DC power line communication network for health sensing and rehabilitation. Their system provides a network through which multiple sensors and actuators can send and receive both information and power. Their DC power-line communication method applies to wires or conductive yarns with high conductivity. Implementation of a near-field coupling transceiver integrated, with a fault-tolerant network switch for inter-layer and intra-layer wearable BAN, is introduced by Yoo and Lee [12]. The proposed method is applicable to multilayer clothes. Because of washing, scratching and stretching of the textiles, the conductivity of the yarns changes over time. Reported results indicate that conductivity of yarns decreases after washing [13]. The conductivity variation may affect the signals in physical layer and should be taken in consideration in the design phase.

Most of published designs for BANs are based on star or bus topologies. Given the trend for higher number of sensors and more complex networks, the use of these topologies will present increasing efficiency problems, making mesh networks a suitable alternative. The hardware design for this particular application of mesh networks should consider the specific characteristics of BANs. The present paper describes the physical and MAC layers of a 4-port PHY module for mesh networks in wearable applications. All links between sensor nodes are bidirectional: each node can send data to a neighboring node and receive data from another neighbor, at the same time. In fact, the transmitter and receiver are independent. All the PHY, except the line drivers (to protect the FPGA) and MAC, is implemented in a low power FPGA. Results obtained with the prototype indicate that the proposed circuit works with both normal wires and conductive yarns, in a wide range of electrical characteristics.

III. Description of the Physical and MAC Layers

This section describes the physical and MAC layers of the yarn-based mesh network. In mesh networking, nodes are connected to each other via radio links (in wireless systems) or point-to-point physical connections (in wired networks). Nodes collaborate to propagate the data through the network. The circuit described in this work is a multiport mesh network device for communication over conductive yarns.

A. MAC Layer and Framing

The interconnecting wires between the nodes are used for shared bidirectional communication. Sharing the communication medium implies, however, a MAC protocol to manage the access. Figure 1 shows the frames and the MAC protocol used to control the transmission. The shaded portion indicates that the line is free (high impedance). As can be seen in Fig. 1-a and -b, the frames are formatted differently, depending on whether data or MAC control frames are transmitted. Dissimilar start bits are used to distinguish the type of the transmitted frame. Because the communication at the MAC layer is node-to-node over dedicated lines between the SNs, there is no need to support explicit addressing in the MAC layer. For the application described in this paper, the addressing method is specified at the network layer [14]. Data frames carry any type of data packets with maximum 256 bytes length, and are protected by a CRC-8 checksum [15]. One-byte MAC messages include two parts: Upper nibble is the message and lower nibble is the CRC-4 [16] of the upper nibble. The CRC polynomials are:

$$\text{CRC-8 polynomial:} \qquad x^8 + x^5 + x^4 + 1$$

$$\text{CRC-4 polynomial:} \qquad x^4 + x^1 + 1$$

Non-return-to-zero inverted (NRZI) mapping is employed for data line coding. The signal level used for transmission with the NRZI method changes only if the bit being transmitted is a logical 1. Unlike what happens with self-synchronizing coding techniques, such as Manchester coding, which require the line bandwidth to be higher than the data rate, the maximum frequency used with the NRZI method is equal to the data rate. Using higher frequencies would increase power consumption and, for a fixed bandwidth, decrease the maximum data rate. It should be noted that all the nodes communicate in an asynchronous mode, meaning that each node must extract the clock signal from incoming signals. However, a stream of zeros is costly in this mode of operation. If the signal is kept constant for long periods of time, the receiver will not be able to recover the clock due to insufficient information (signal transitions). Therefore, a 1 bit is added after each byte of the data stream to ensure that at least one signal level transition occurs for every 8 bits of data. This synchronization bit is also used to define the end of a frame, by changing its value to 0.

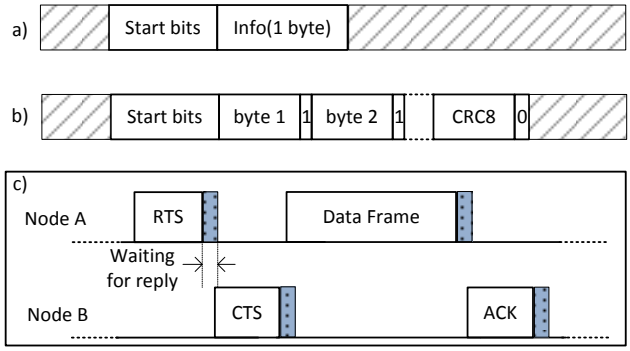

Fig. 1. Media access layer: a) MAC control frame; b) MAC data frame; c) MAC protocol.

Communication in the MAC layer is controlled by the RTS/CTS (Request-To-Send/Clear-To-Send) handshaking mechanism presented in Fig.1-c. Since BANs need to transmit data acquired in real-time, there is a need for reliable communications [17]. In mesh networks each node must be able to handle communication requests from different neighbors simultaneously. In order to efficiently manage the requests, sender and receiver nodes should be aware of the status of each other. Otherwise, a significant number of data packets may be lost and require retransmission. The use of RTS/CTS

handshaking helps to ensure reliable communication with low packet loss and good resource management.

The sender starts the communication with an RTS (Request-To-Send) message. If the receiver node is ready to receive the packet, it replies with a CTS (Clear-To-Send) message over the same line. Then, the sender dispatches a packet encapsulated in a data frame. If the packet is checked as error-free, the communication ends with an ACK message generated by receiver; otherwise an ERR message is sent to the peer node. After each message, the sender node waits for a reply from the receiver for a limited amount of time. The waiting time corresponds to the time required to transmit 8 bits (for ACK, CTS, and ERR), or 16 to 32 bits (for RTS). Lack of reply within the specified time indicates a communication error. For the RTS/CTS handshake, if the sender node does not receive a CTS message in time, it repeats the RTS message. The number of tries is configurable; RTS messages can be repeated up to 255 times before the receiver is considered unavailable. If two nodes start to send RTS messages simultaneously, a collision will occur. To avoid the repetition of collisions, each node send RTS messages with randomly selected delays (with a duration corresponding to between 16 and 32 bits).

B. Organization of the PHY Module

Fig. 2 shows the block diagram of the PHY module. The acronym TX represents the transmitter module. It includes all the sub-modules needed for buffering, encoding, generating and detecting MAC information, and also for frame generation. Conversely, label RX indicates the receiver. This module includes all sub-modules needed for decoding, generating and detecting MAC frames, buffering of packets carried in data frames and for error checking. The line control module is responsible for detecting signals and for switching the lines to receive or to transmit the packets. The Clock module generates the internal clock, resets the circuit and sets the bit rate. All the setting and data reading from RX-FIFO, or writing to the TX-FIFO, is controlled by the Control module via an internal bus and a SPI port. The SPI port provides a high-speed communication channel to the microcontroller.

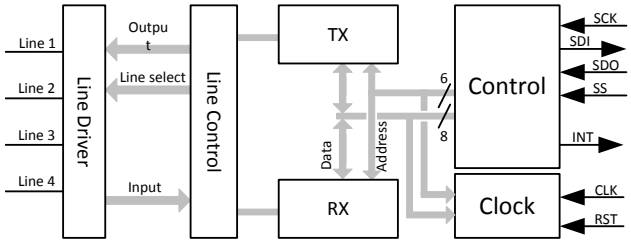

Fig. 2. Overall block diagram of the network device.

Figure 3 shows mote details about the organization of the TX module that controls transmission of data. Before starting any communication, TX checks if the line is available using the *Line_Status* signals generated by the Control module. FIFO buffers of 256 bytes size, needed for data storage before sending, are implemented with the FPGA's internal memory. The last byte of data in the frame is a CRC-8 checksum used for error checking during communication, and is generated by the TX module. Before starting transmission, the TX control registers of the TX_Control module must be initialized. The TX control registers are:

- *TX_Status*—determines when to start sending data;
- *PktLength*—sets the data size in the FIFO to be sent;
- *Repeat_RTS*—defines the number of RTS messages to send in case of no reply from the receiver;
- *TX_Line*—appoints which line will be active for sending the data.

Depending on the outcome of the transmission, the TX_Control module generates output signals to flag the status of transmission.

The TX_Timing module is responsible for setting the clock of TX module. It is controlled by the TX_control central unit, and defines the overall TX module clock rate. To minimize power consumption, TX_Control turns TX_Timing off when no data is available for transmission. The Clock module generates the circuit timing, setting both the system clock and bit rate. The encoder unit includes all sub-modules for data encoding and generation of the synchronization bits. It generates the MAC messages and determines the end of transmission. The input data is serial, supplied by a PISO (parallel-in serial-out) module that serializes the FIFO output. The output of the encoder is then connected to the NRZI line encoder.

When a node wishes to start communication, it must be able to detect MAC messages. The TX module has the ability to process them independently from RX, enabled by the MAC_receive unit built within TX. It actually allows the PHY module to simultaneously send and receive data from different lines.

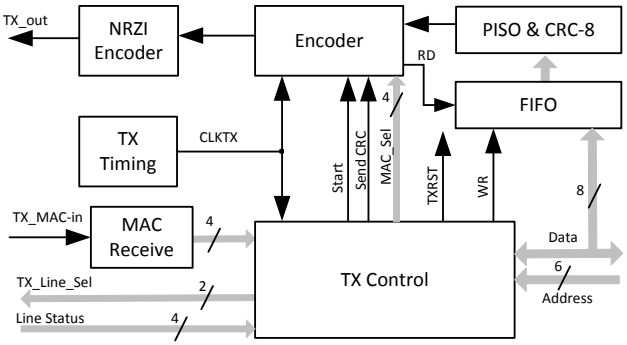

Fig. 3. Block diagram of the transmitter module.

The RX module shown in Fig. 4 is responsible for receiving packets. Like the TX module, received data is accumulated in FIFO buffers and read via an SPI port. RX_Control is connected to the internal bus and includes the receiving control and registers.

To reduce power consumption, RX_Control turns off the clock signals in all RX modules that are not active. Activa-

tion starts when the signal detector module detects an RTS message. After reception of the data frame, and according to the CRC-8_Check module output, RX_Control sets the receiver status according to the output of the CRC-8_Check module. RX_Timing recovers the synchronization clock from the incoming data signals. The RX_Timing unit also detects the frame start bits and sends them to the Decoder. It decodes the NRZI data, detects the end bit and MAC messages, and converts decoded serial data into parallel by means of an SIPO (serial-in parallel-out) unit.

The CRC-8_Check module checks the received data for errors. The output is validated whenever the Decoder module detects the end of the frame. If CRC-8_check indicates an error, RX_Control sends an ERR message to the corresponding transmitter node. Similarly to the TX, the RX module also has a simple transmitter built within to autonomously send MAC messages.

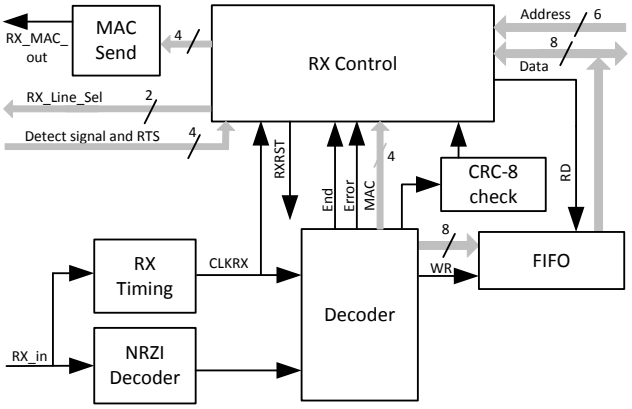

Fig. 4. Block diagram of the receiver module.

The control of the line status and the switching of the lines between RX and TX modules are the tasks of the Line_control unit shown in Fig. 5. This module includes multiplexers, demultiplexers, together with the line_status and signal_detector submodules. When a line is in free mode, the signal_detector detects the RTS message composed of a stream of ”1”s. Line_staus module indicates if a line is free or occupied. When occupied, the signal_detector module deactivates the detector on that line and the TX module stays in a waiting mode until the line is free. Two multiplexers, at the input port, multiplex the signal to RX or MAC_receive module in the TX module. The TX_Line_Switch includes the demultiplexers that switches the TX_out and MAC_send signals to the selected output. The Line_Control is connected to Line_driver, which interfaces with the conductive.

IV. The Implemented Prototype

This section presents a prototype for sensor nodes to evaluate the proposed organization. This prototype was designed for a particular BAN application devoted to the capture of muscle electromyographic signals and movement information from the lower limbs. Fig. 6 shows a diagram of the prototype. It includes a 16-bit microcontroller and one FPGA (Actel

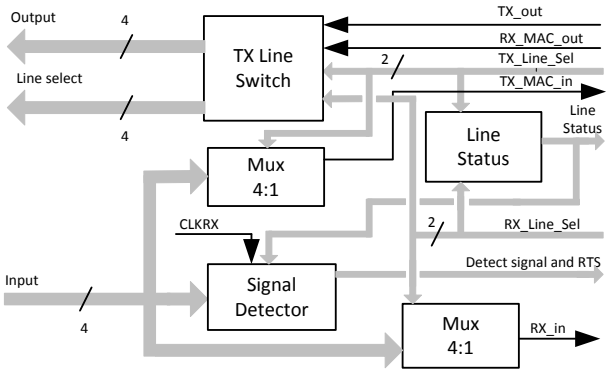

Fig. 5. Block diagram of the line control module.

AGLN125 [18]) on the main PCB board; sensors and line drivers are on separate PCB boards. The FPGA model was selected because of its low power consumption. The power supply is a coin type battery. A DC-to-DC converter generates a 3,3 V power supply voltage for the microcontroller and sensors, and a 1,5 V supply voltage for the FPGA and line driver circuits.

The internal oscillator of the microcontroller (16 MHz) is the clock source for both microcontroller and FPGA. The microcontroller implements the networking and application layers, acquires the signals from EMG sensors and kinematic data from accelerometers and gyroscopes. The code was written in C language with MPLAB X IDE tools and the microcontroller was programmed with the PICkit 3 programmer. FPGA setting registers are available for reading and writing by the microcontroller via an SPI port.

The hardware implementation of the PHY and MAC layers, presented in Section III, have been described in Verilog language, synthesized with in Libero Project Manager V9.1 from Actel, and used to program the FPGA with the FlashPro4 device programmer.

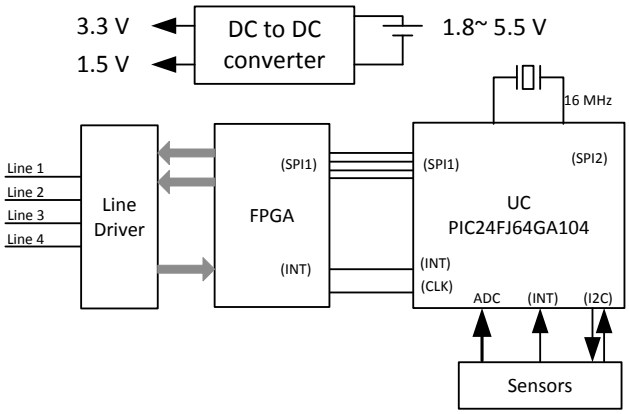

Fig. 6. Overall organization of the communication node prototype.

Figure 7 shows the main board (2 layers, $6 \times 6\,\mathrm{cm}^2$) with a line driver (upper right corner) which is connected to the main board by a 20-pin connector (not visible). Supply voltage,

clock frequency, and FPGA utilization are summarized in table I .

TABLE I
CHARACTERISTICS OF THE FPGA IMPLEMENTATION

Parameter	Value
Model	Actel IGLOO AGLN125
Supply voltage	3,3 V
Clock frequency	16 MHz
Logic cell utilization	2509 of 3072 (81,7 %)
Internal memory	512 bytes

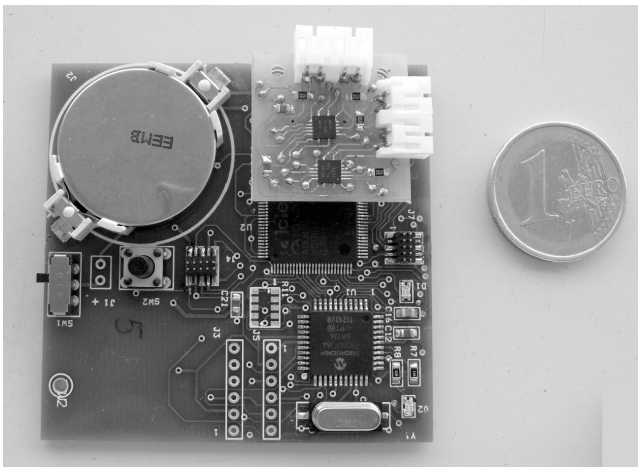

Fig. 7. Picture of the communication node prototype.

V. EXPERIMENTAL RESULTS

The results presented in this section were obtained with the prototype just described. For the purposes of performance analysis, the communication nodes were connected together with normal twisted wires. The same measurements were repeated with conductive yarns, with the former test, with twisted wires, acting as a control experiment. Signals flowing in the communication lines have been observed and measured with an oscilloscope.

Figure 8 shows the resulting signal when two sensors are connected with normal twisted wire and communicating at a 2 Mbps rate. It can be seen that, in accordance with the MAC protocol, communication lines are shared among the transmitter and receiver. The signal level at the output is between 0 V and 1,5 V.

Figure 9 shows the signals measured at the RX port, but now at 9 Mbps rate and with 256 bytes of data (maximum packet size). In this measurement, one of the wires is a conductive yarn with 50 Ω resistance and 60 cm length in the relaxed state, or 120 cm in the stretched state. The circuit is capable of detecting the signals, even for those with degraded rising or falling edges. The measurement was repeated in a loop from the transmitter output to another free port in the same sensor node. The circuit was able to send and receive packets simultaneously. It is also capable of realizing 9+9 Mbps data transmission, either over a conductive yarn or a normal wire. This is a higher bit rate than those reported in [12] (10 Mbps), [7] (256 kbps) or [19] (100 kbps).

Next, a 2,2 kΩ resistor was placed in series with the communication lines. Even at a bit rate of 9 Mbps the receiver is capable of detecting and correctly decode the packet, although the receiving signals are of dubious quality, as Figure 10 indicates. As mentioned before, conductive yarns display electrical properties variations in relaxed and stretched modes; in addition their conductivity changes over time. Experimentally measured electrical parameters for the wires used in the Prolimb project indicate that the yarns mainly have resistive behaviour [20]; a typical value for one specific type of conductive yarn is 6,75 Ω/cm. The results for the two different series resistors show the ability of the circuit to perform correctly and maintain performance levels over the whole conductivity range of typical yarns.

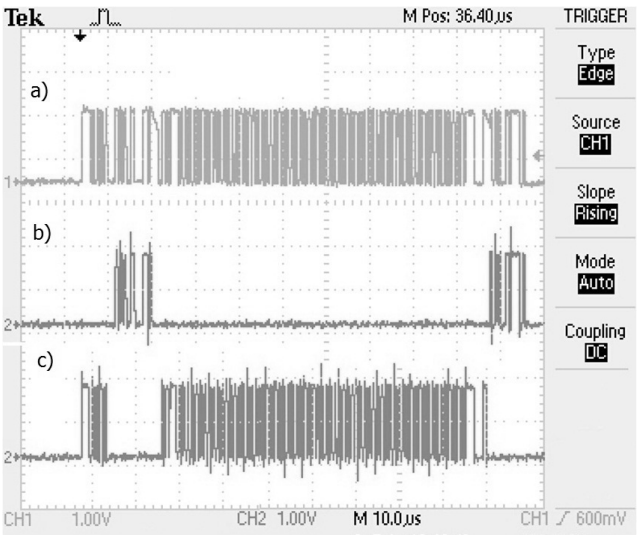

Fig. 8. Observed signals: a) Signals on the line, b) CTS and ACK signals genereted by RX, c) RTS and data frame from TX.

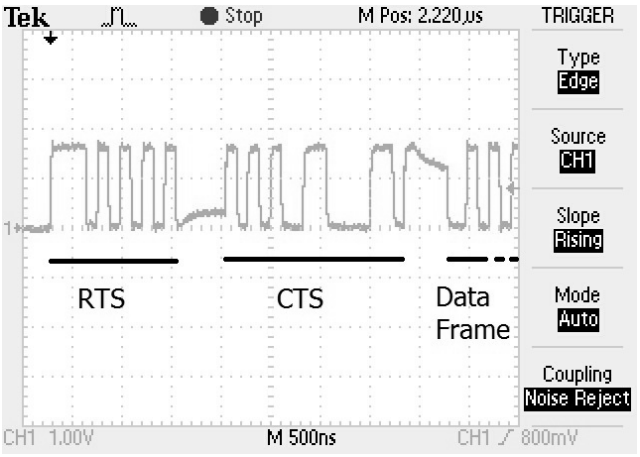

Fig. 9. Observed signals with conductive yarn in stretched mode.

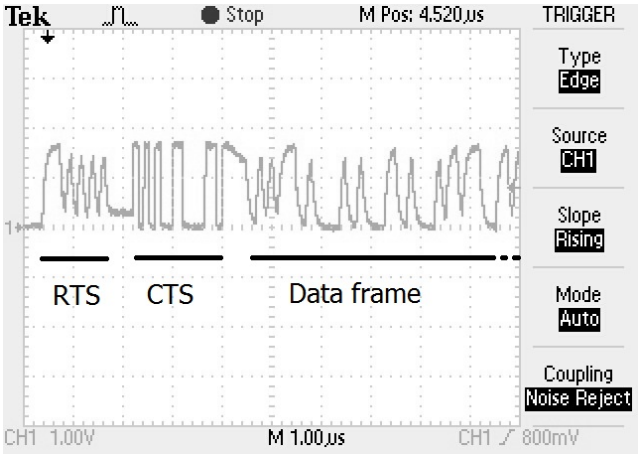

Fig. 10. Observed signals with a 2,2 kΩ resistor in series with line at 9 Mbps data rate (256 byte packet).

The measurement of power consumption in idle mode indicates that the power expenditure for data rates below 512 kbps is 1,5 mW. It increases to 1,74 mW at 1 Mbps, 1,95 mW at 2 Mbps, and 2,7 mW at 4 Mbps. The increased power consumption in idle mode is due to the signal detector module, whose clock signal must be increased according to the data rate even in idle mode.

In order to operate at data rates above 4 Mbps the internal PLL of FPGA needs to be activated, which increases the power consumption. To attain the maximum data rate (9 Mbps), the PLL output frequency must be set 70,993 MHz, increasing the power consumption to 7,92 mW.

The power consumption during communication increases to 3,5 mW (at 4 Mbps) and 9 mW (at 9 Mbps) in TX mode; in RX mode the power consumption is 4 mW (at 4 Mbps) a 10 mW (at 9 Mbps). Usually wired system consume less power than wireless systems. The values reported value in [1] for Zigbee (30 mW, 250 kbps) and Bluetooth (100 mW, 3 Mbps) indicate that the proposed circuit consumes substantially less power than wireless alternatives. The current prototype is implemented on FPGA; a future ASIC implemention of the architecture will show even better performance.

VI. Conclusion

This paper describes a network device for low-power wearable BAN applications. The device is targeted at networks of sensors connected in a mesh topology by conductive yarns, and supports full-duplex communication. Design and implementation of the physical and MAC layers, and a hardware prototype have been described. The measurement results obtained from the prototype show that the proposed circuit can work up to 9+9 Mbps. The measured speed is high enough to satisfy most of BAN application requirements. The ability of the circuit to detect low quality signals in the communication lines, low-power operation and high data rate make it a suitable system for use in wearable applications with conductive yarns.

Acknowledgments

This work is partially funded by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) – within project PROLIMB (PTDC/EEA-ELC/103683/2008) and through grant SFRH/BD/75324/201.

References

[1] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *IEEE Trans. on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 40, no. 1, pp. 1–12, Jan. 2010.

[2] A. Lymberis and L. Gatzoulis, "Wearable health systems: from smart technologies to real applications," in *28th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2006. EMBS '06*, vol. Supplement. IEEE, Sep. 2006, pp. 6789–6792.

[3] L. Gatzoulis and I. Iakovidis, "Wearable and portable eHealth systems," *IEEE Eng. in Medicine and Biology Magazine*, vol. 26, no. 5, pp. 51–56, Oct. 2007.

[4] J. Xing and Y. Zhu, "A survey on body area network," in *5th International Conference on Wireless Communications, Networking and Mobile Computing*. IEEE, Sep. 2009, pp. 1–4.

[5] I. Howitt and J. A. Gutierrez, "IEEE 802.15.4 low rate - wireless personal area network coexistence issues," in *IEEE Wireless Comm. Network.*, vol. 3. IEEE, Mar. 2003, pp. 1481–1486 vol.3.

[6] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. C. Leung, "Body area networks: A survey," *Mob. Netw. Appl.*, vol. 16, no. 2, pp. 171–193, Apr. 2011.

[7] Z. Nakad, M. Jones, and T. Martin, "Fault tolerant networks for electronic textiles," in *2004 Intl. Conf. Comm. in Computing*, 2004, pp. 100–106.

[8] H. S. Lee, "Wearable personal network based on fabric serial bus using electrically conductive yarn," *ETRI Journal*, vol. 32, no. 5, pp. 713–721, Oct. 2010.

[9] F. Mizuno, T. Hayasaka, K. Tsubota, S. Wada, and T. Yamaguchi, "Development of a wearable computer system with a hands-free operation interface for the use of home health caregiver," *Technology and Health Care*, vol. 13, no. 4, pp. 293–300, 2005.

[10] E. R. Post and M. Orth, "Smart fabric, or wearable clothing," in *Digest of Papers First Intl. Symp. Wearable Computers*. IEEE, Oct. 1997, pp. 167–168.

[11] E. Wade and H. H. Asada, "Wearable DC powerline communication network using conductive fabrics," in *Proc. IEEE Intl. Conf. Robotics and Automation*, vol. 4. IEEE, May 2004, pp. 4085–4090 Vol.4.

[12] J. Yoo, S. Lee, and H. Yoo, "A 1.12 pJ/b inductive transceiver with a Fault-Tolerant network switch for Multi-Layer wearable body area network applications," *IEEE Journal of Solid-State Circuits*, vol. 44, no. 11, pp. 2999–3010, Nov. 2009.

[13] S. Varnaite and J. Katunskis, "Influence of washing on the electric charge decay of fabrics with conductive yarns," *FIBRES & TEXTILES in Eastern Europe*, vol. 17, no. 5, pp. 69–75, 2009.

[14] F. Derogarian, J. C. Ferreira, and V. M. G. Tavares, "A routing protocol for WSN based on the implemention of source routing for minumum cost forwarding method," in *Proc. Fifth Intl. Conf. Sensor Tech. Appl. (SENSORCOMM 2011)*, J. M. Hovem, J. Ellul, L. Gomez, and Y. Reddy, Eds. ThinkMind, Aug. 2010, pp. 85–90.

[15] *Understanding and Using Cyclic Redundancy Checks with Maxim iButton Products*, Dallas Semiconductor, Mar. 2001, app. note 27.

[16] *ITU-T Recommendation G.704*, International Telecommunication Union, 1998.

[17] M. Chen, S. Gonzalez, A. Vasilakos, H. Cao, and V. C. M. Leung, "Body area networks: A survey," *Mobile Networks and Applications*, vol. 16, no. 2, pp. 171–193, 2010.

[18] *IGLOO nano FPGA User's Guide*, Microsemi Corporation, Dec. 2011.

[19] I. Locher, H. Junker, T. Kirstein, and G. Troster, "Wireless, low-cost interface for body area networks," in *Eighth Intl. Symp. Wearable Computers*, vol. 1. IEEE, Nov. 2004, pp. 170–171.

[20] A. Zambrano, F. Derogarian, R. Dias, M. J. Abreu, A. Catarino, A. M. Rocha, J. M. da Silva, J. C. Ferreira, V. M. G. Tavares, and M. V. Correia, "A wearable sensor network for human locomotion data capture," in *Proc. 9th Intl. Conf. Wearable Micro and Nano Technologies for Personalized Health*, Jun. 2012.