A Lightweight communication protocol for Body Area Networks

Juraj Ďuďák
Slovak University of Technology
Faculty of Materials Science and Technology in Trnava
Jana Bottu 2781/25, Trnava, Slovakia and
University of Zilina, Research center
Univerzitná 8215/1, Žilina, Slovakia
juraj.dudak@stuba.sk

Jakub Perička
Slovak University of Technology
Faculty of Materials Science and Technology in Trnava
Jana Bottu 2781/25, Trnava, Slovakia
jakub.pericka@stuba.sk

Martin Skovajsa
University of Zilina
Research center
Univerzitná 8215/1, Žilina, Slovakia
martin.skovajsa@uniza.sk

Abstract. *This article introduces nanoBus (nBus), an innovative RS485-based communication protocol designed specifically for Body Area Networks (BANs). To enhance real-time monitoring and feedback of health parameters, nBus addresses the growing need for reliable and secure patient data transmission, especially in clinical and home rehabilitation contexts. Utilizing the RS-485 serial interface, nBus ensures high data transfer rates and robust resistance to interference, a notable improvement over wireless technologies in terms of security and reliability. The protocol operates on a master/slave request/response model, facilitating node and sensor addressing within a BAN. The architecture of nBus is thoroughly detailed, highlighting its three primary layers: Physical, Link, and Application. This comprehensive design allows for scalable integration of up to 132 nodes with 32 sensors each, promising significant advancements in BAN communication. The paper concludes with prospects, emphasizing the protocol's potential implementation in intelligent clothing and the challenges therein, particularly in cable management.*

Keywords. nBus Protocol, RS-485 Communication, Body Area Network, Health Monitoring, Wearable Sensors

1 Introduction

As the World Health Organization notes, the number of aging people and patients with chronic diseases, obesity, and other disabilities is steadily increasing and becoming an unavoidable problem. It will likely contribute to a significant deterioration in the quality of care in hospitals and health centers as more patients are readmitted daily (Rahat, 2018). The need for optimal and rapid healthcare solutions to tackle the looming health crisis is the focus of many researchers (Ullahm, 2012).

In outpatient practice, many vital functions (heart rate, blood pressure, body temperature, and similar) are monitored. However, they are rarely monitored simultaneously, often resulting in an incomplete picture of the patient's health status. To solve this problem, non-intrusive and real-time ambulatory health monitoring offers an efficient and cost-effective way to treat patients, as vital signs can be recorded and medical data updated in real-time (Ullahm, 2012).

Small sensor devices that can monitor various patient data have become increasingly popular in the healthcare industry in recent decades. It is mainly due to advances in nanotechnology, which have enabled the development of nano biosensors with high sensitivity and a range of functions for monitoring and diagnosis even at the cellular level (Javaid, 2022). This technological progress is particularly noticeable in the age of the Internet of Things (IoT), big data, and cloud computing. In terms of human health monitoring, the Body Area Network (BAN), also known as the Body Sensor Network (BSN), seems to be a prime candidate among the most promising communication technologies (Ullahm, 2012).

These networks consist of small, low-power, and intelligent wearable sensors that are placed in, on, or around the human body. These sensors constantly collect data on the health status and activities of the person and store it for further processing (Rahat, 2018; Ullahm, 2012). As part of these networks, multiple actuators (displays, speakers, buzzers, and others) can also be implemented to provide feedback to the user in an interactive way (Zhang, 2009). The communication hierarchy in BANs is generally divided into three parts, as shown in Fig. 1:

- *Intra-BAN*, which includes communication between sensors, actuators, and a control unit,
- *Inter-BAN* consisting of personal devices and other BAN instances,
- *Beyond-BAN* relates to data transmission over the WAN, where it can be stored, analyzed, and action taken in an emergency.

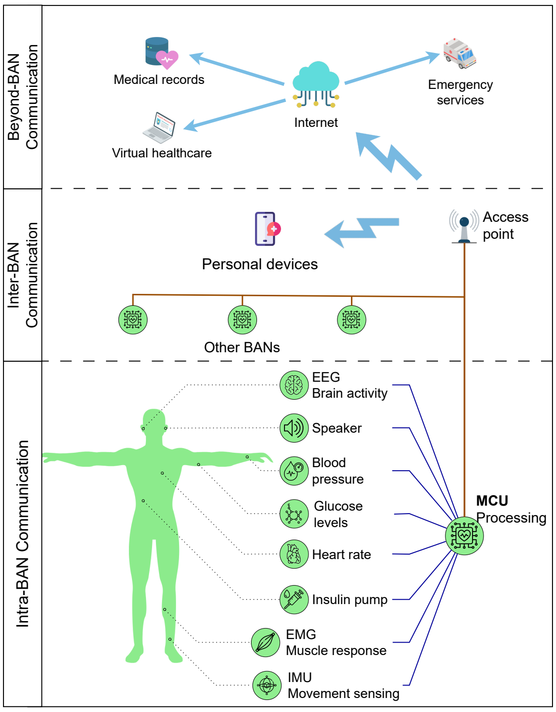

Figure 1: BAN Communication hierarchy

1.1 Related Work

Recently, the main focus has been put on the development of new communication protocols for intra-BAN communication, especially for wireless applications. Wireless technologies (IEEE 802.15.6, Bluetooth, ZigBee, and others) have many advantages. The most important one is mobility, as the central concept of a BAN is that it cannot restrict the user in any way. There are also low-power technologies that allow battery-powered devices to last a long time. However, the protocols based on wireless sensor networks (WSN), which consist of stationary and redundant sensor nodes, are not often reliable in a BAN environment with much movement and suffer from interference, packet loss, and latency issues.

Using air as a transmission medium also raises the issue of security, especially as patient data is susceptible. These challenges put researchers and manufacturers alike under considerable pressure (Shadi, 2021). In this detailed study, the author deals with the placement of sensors and the human body, the possibilities of their mutual communication.

Patel in his paper (Patel, 2010) lists appropriate technologies, minimum transmission rates, minimum latency, and required uptime as parameters to use for tracking the desired biosignal. He also considers QoS and communication security as a key factor in selecting the appropriate technology. All the technologies described are exclusively wireless.

In the paper (Li, 2008), the authors describe the standardization of BANs according to the IEEE 802.15.BAN standard. Low power consumption, guaranteed response to external stimulus were defined as key parameters in the implementation. As transmission medium they define: air, vicinity of human body, inside human body. The transmission rate is defined from 100 bps (heartbeat, blood pressure, body temperature) up to 2500 bps (Electro cardiogram).

The proposed nBus protocol aims to meet the requirements of high transmission rate (up to 2 Mbps), wireless link, higher number of connected sensors (up to 32 sensors), transmission reliability and energy efficiency.

2 NanoBus (nBus) Protocol

The architecture of nBus is thoroughly detailed, highlighting its three primary layers: Physical, Link, and Application. The Physical Layer emphasizes using RS-485 for its industrial-grade reliability and high data rates, which are suitable for short BAN bus lengths. The Link Layer focuses on efficient node addressing and error checking, including an 8-bit cyclic redundancy check (CRC-8). The Application Layer encompasses the translation of requests and responses, defining up to 128 function codes for various operations.

The nBus protocol is an application layer protocol for connecting sensor nodes to a higher-level control unit in the BAN. The main development goal is to provide a reliable and secure way for real-time monitoring and feedback of health parameters during clinical or home rehabilitation and physiotherapy. The communication model of nBus is shown in Fig. 2.

2.1 Physical Layer

The communication is a master/slave request/response model, allowing node addressing and specific sensor addressing, as a communication node might consist of sensors and actuators. The packet structure consists of a link frame and an application packet as shown in Fig. 3.

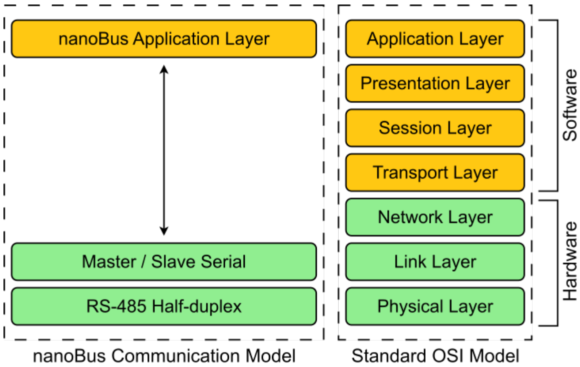

Figure 2: Communication Layer Model

byte.0	byte.1	byte.2	byte.3	byte.4	byte.5
Length	Node address	Sensor address	Function	Data	CRC-8

Data Link Frame
nBus Data Packet

Figure 3: nBus packet structure

To achieve reliable and secure data transmission between the nodes, we have opted for an RS-485 serial interface, an industry standard with high reliability and resistance to interference. Since the transmission is via cable, it is more tap-proof than wireless technologies. RS-485 is a half-duplex mode that uses only two twisted differential wires for data transmission, often combined with a common ground wire in a single cable. It was developed for connections in harsh industrial environments, enabling high data rates over long distances.

The standard states that a rate of 100 kbit can be achieved with a bus length of approx. 1200 m (Texas Instruments, 2008) with a maximum of 32 nodes on one bus due to current loads. However, as the bus of a BAN is very short ($< 5\ m$), even 100 Mbit rates are achievable if high-end transceivers are used. These have a 1/5 bus load, meaning up to 160 nodes can be connected to a single bus without repeaters. In addition, modern integrated circuits can be put into a low-power standby mode when no transmission occurs, extending battery life for portable devices (Renesas, 2017).

A crucial thing to consider is cable management. The data rates for high-speed transmission use a CAT5 cable as a reference. The ANSI/TIA/EIA-568-B standard dictates a minimum wire gauge of 26 American Wire Gauges (AWG) or 0.14 mm^2 (ANSI/TIA/EIA, 2001). By using thin 3-wire cables, we could reduce the impact on patient's mobility and increase the potential for smart clothing use. However, it also poses new challenges, with maintenance being the prime candidate as thin wires are prone to breaking with repeated movement.

2.2 Link Layer

The link layer is responsible for node addressing and error checking. Each node in a BAN has its unique node address. Each frame contains an 8-bit cyclic redundancy check (CRC-8) to implement error detection during transmission. This check is the last byte transmitted, and to account for the dynamic frame length, the frame length is included as the first byte. The maximum frame length is currently set to 132 bytes.

2.3 Application Layer

The application layer is responsible for the translation of requests and responses. It works either in slave or master mode. In a single nBus network, only one master is permitted, which sends data requests, slave commands, and application error handling. The information about the target service is contained in a single function byte and supports up to 128 function codes. The MSB of the function byte is reserved for an error flag.

In the following terminology, the following terms are used: master node — a device connected to the BAN that communicates with slave node devices, slave node — a device in the BAN to which sensors sensing signals of interest are connected, sensor — a console sensing element, a single slave node may contain multiple sensors. Each sensor can be addressed individually or as a group.

These functions range from version information, data request, node, and sensor parameter configuration to node echo, calibration, sleep, and wake requests. nBus supports broadcast for the entire network (e.g., clock synchronization across nodes or calibration of all inertial sensors), broadcast for individual node sensors (e.g., setting the same gain for each force sensor in a single node), and also node sensor unicast (i.e., sending data for display on an integrated display).

bit.7	bit.6	bit.5	bit.4	bit.3	bit.2	bit.1	bit.0	
Type			Sensor address					group
$I/\bar{O}$	Res	Res	a_4	a_3	a_2	a_1	a_0	bit

Figure 4: Sensor Address byte structure

The target sensor is specified in the node address byte as shown in Fig. 4. The first three bits are used to indicate the sensor type, with the first bit's designation as an input/output selector and two reserved for later versions. The remaining bits are used for sensor addressing. This way, we can integrate up to 31 sensors and actuators into a single node.

The slave node listens to the master's requests and generates corresponding responses. The distinction between transmission types is made by node and sensor address bytes, as shown in Tab. 1 where $X \in (1,\ 127)$ represents a node address and Y being an sensor address ($Y \in (1,\ 31)$ for outputs, $Y \in (129,\ 159)$ for inputs). Both slave and master nodes implement transmission error detection and correction. In the event of a transmission error, the node that detects the error sends a data packet with a set error flag and an error code as data. This code can be anything from a bad CRC result to an invalid parameter request.

Table 1: nBus address bytes based on transmission type

Node Address	Sensor Address	Description
0	0	Broadcast to all nodes and sensors.
0	Y	Broadcast to specific sensors.
X	0	Unicast to slave node (to all node sensors).
X	Y	Unicast to node sensor.

3 nBus Application Protocol

The nBus protocol was designed with an emphasis on minimizing communication overhead by allowing the use of high data rates and advanced addressing options (unicast/broadcast/group broadcast). This protocol is designed as a library independent of the hardware platform used. In the current version, this library is implemented for the STM32L031 (Slave module) and Nordic NRF52820 (Master module) hardware platforms. The nBus library has been programmed in C, which allows it to be used in both C and C++ projects. The control firmware was programmed in the STM32CubeIDE development environment (the STM32L01x microcontroller was used) and Visual Studio Code/Platformio (for the NRF52820 microcontroller).

For easy implementation of the nBus protocol on the selected hardware platform, the *nBusPlatformInterface_t* interface is available to provide a connection to the platform on which the library is implemented. This interface consists of two layers containing elementary functions. These functions need to be implemented on a specific microcontroller:

- **Communication layer**. Receiving and sending data packets:
  - `uart_receive()` — receiving a data packet in non-blocking mode,
  - `uart_transmit()` — sending a response from the application layer of the nBus protocol.
- **System Layer**. Notification of the status of connected sensors:
  - `loop_callback()` — nBus application layer notification of external events,
  - `delay_ms()` — blocking delay.

The second part of the slave module functionality consists of the connected sensors. For a uniform implementation of the connected sensors in the slave module, the *nBusAppInterface_t* interface has been prepared to provide uniform access to the data and settings of the left-handed sensor connected to the module. This interface provides the following functionality:

- initialization and reset of sensor settings,
- obtaining information about the number and type of connected sensors within the hardware module,
- setting and reading the configuration parameters of the connected sensors,
- start and stop the automatic sensor reading process,
- read the value from the sensor(s) of the module.

These 2 interfaces ensure the independence of the hardware platform used. When you need to connect a module with a new type of sensors to the BAN, you only need to implement the *nBusAppInterface_t* interface.

Each module must store the configuration of the connected sensors in non-volatile memory. The internal design of the memory map was designed to be able to store the configuration of 16 sensors, each sensor can have up to 8 parameters. The parameters were chosen to suit the widest possible use (time base, resolution, gain, offset, sample rate, range 1, range 2, filter). The use of these settings is optional. The *nBusAppInterface_t* interface, which implements the functionality for each existing sensor, must indicate whether a given parameter is implemented for a particular sensor.

3.1 Sensoric module nBus slave

In the actual version, we have prototyped an nBus module with a 6-axis IMU sensor. The module is built on the STM32L0130 MCU — Fig. 5.

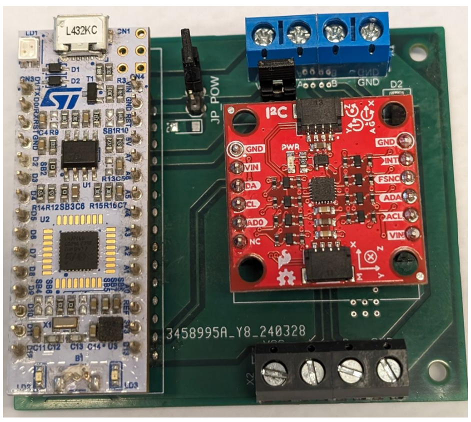

Figure 5: nBus module with IMU sensor

Properties of nBus slave module on Fig. 5:

1. communication interface RS485, with baud rate up to 2 Mbps,
2. Connected sensors: 1 * accelerometer, 1 * gyroscope (in three axes),
3. maximum frequency of IMU sensor readings: 260 Hz,
4. it contains internal non-volatile memory for storing the configuration of the module itself and all connected sensors,
5. device does not contain a battery, the power supply of the module is solved via RS-485 bus.

An important aspect of the overall design is the ability to ensure data transmission at the required quality. In the context of a BAN, we consider bus transmission capacity as one of the main quality indicators. According to the above proposal, a BAN can have up to 127 nodes, with each node having up to 31 sensors. Thus, in total, the network can contain up to 3937 sensors.

Our design includes 2 network interfaces: an RS-485 interface for the internal network and an RF interface for communication with the external system. The communication in the internal network must be conflict-free and with maximum speed. This is ensured by the master/client architecture.

$$C = (N * Data_{size} + C_{overhead}) * f \tag{1}$$

To calculate the required baud rate of the internal bus, we use equation 1.

Where C is the required communication capacity of the bus for a single slave module. N is the number of active sensors, $Data_{size}$ is the word width (number of bytes per data), $C_{overhead}$ is the communication overhead and f is the communication frequency. We consider with N=31, $Data_{size} = 2$, $C_{overhead} = 10B$ (request length and overhead in the slave module response) and $f = 100Hz$.

$$C = (31 * 2B + 10B) * 100Hz = 7200B/s \tag{2}$$

For a theoretical number of modules n=127, we get the required capacity of the communication link $C = 914400B/s$.

3.2 Future work

The current aim of the development is to implement the nBus protocol in embedded applications, with the target processing unit being low-power 32-bit microprocessors. A hardware platform consisting of a high-speed RS-485 interface, analog-to-digital converters, a USB debugging interface and an STM32L031 microprocessor from STMicroelectronics is being developed for testing purposes. Power management is also an important part of the overall design. Setting the minimum power consumption of the nBus slave is determined by using the appropriate operating modes of the microcontroller, but also by taking advantage of the sleep modes of the sensors used. Another criterion is the actual mechanical design of the whole module. As this module will be attached to a human limb, the overall weight needs to be minimized so that the weight of the device itself does not affect the measurement results.

We are currently working on the nBus master module, which will additionally contain a battery to power the entire nBus network and a hardware interface for wireless transmission of measured data from the nBus slave modules.

4 Conclusion

In this article, a new communication protocol for Body Area Networks has been proposed. The main goal of the design is to develop a reliable and straightforward protocol for real-time monitoring of health parameters with high scalability and the ability to implement feedback elements in the system. nBus is an application layer protocol and uses request/response communication between master and slave nodes. An RS-485 industrial bus is the physical layer to ensure high data rates over 2-wire connections. It lightens data security and encryption requirements and, therefore, processing times and energy consumption, as the wired transmission of sensitive patient data is less susceptible to interception. It also opens up new possibilities for implementation in sensor-enabled clothing and poses new challenges in cable management and maintenance.

The nBus data link layer addresses the nodes and checks for transmission errors using CRC-8. Up to 127 nodes are currently supported by the protocol, with each node containing up to 62 sensors (31 inputs and 31 outputs). The central data unit of the protocol is the nBus data packet, which contains the sensor address, function, and data bytes. Further protocol testing still needs to be carried out to obtain objective indicators of the protocol's performance. However, current development focuses on implementing a customized hardware platform based on an STM32L031 32-bit microprocessor (32 kB FLASH, 8 kB RAM, $f_{max} = 32MHz$) and NRF52840 (512 kB FLASH, 128 kB RAM, $f_{max} = 64MHz$)

To summarise, nBus is a novel communication protocol that has the potential to simplify data transmission in BANs and enable reliable communication in real-time applications.

nBus is a pivotal development in the field, offering an efficient and scalable solution for BAN communications, crucial for modern healthcare monitoring systems

Acknowledgments

This work was supported by VEGA through the Research on inertial data analysis methods for applications in rehabilitation adjuvants under Grant 1/0095/24.

References

Min Chen, Sergio Gonzalez, Athanasios Vasilakos, Huasong Cao, and Victor C. M. Leung. (2011) Body area networks: A survey. *Mobile Networks and Applications*, 16(2):171–193. DOI: 10.1007/s11036-010-0260-8.

Sana Ullah, Henry Higgins, Bart Braem, Benoit Latre, Chris Blondia, Ingrid Moerman, Shahnaz Saleem, Ziaur Rahman, and Kyung Sup Kwak. (2012) A comprehensive survey of wireless body area networks. *Journal of Medical Systems*, 36(3):1065–1094. DOI: 10.1007/s10916-010-9571-3.

Shumaila Javaid, Sherali Zeadally, Hamza Fahim, and Bin He. (2022) Medical sensors and their integration in wireless body area networks for pervasive healthcare delivery: A review. *IEEE Sensors Journal*, 22(5):3860–3877.

Khalid Hasan, Kamanashis Biswas, Khandakar Ahmed, Nazmus S. Nafi, and Md Saiful Islam. (2019) A comprehensive review of wireless body area network. *Journal of Network and Computer Applications*, 143:178–198. DOI: 10.1016/j.jnca.2019.06.016.

Rahat Ali Khan and Al-Sakib Khan Pathan. (2018) The state-of-the-art wireless body area sensor networks: A survey. *International Journal of Distributed Sensor Networks*, 14(4):1550147718768994. DOI: 10.1177/1550147718768994.

Mi Zhang and Alexander A. Sawchuk. (2009) A customizable framework of body area sensor network for rehabilitation. In 2009 *2nd International Symposium on Applied Sciences in Biomedical and Communication Technologies*, pages 1–6. DOI: 10.1109/ISABEL.2009.5373703.

Rim Negra, Imen Jemili, and Abdelfettah Belghith. (2016) Wireless body area networks: Applications and technologies. *Procedia Computer Science*, 83:1274–1281. DOI: 10.1016/j.procs.2016.04.266.

Rui Li, Daniel T H Lai, and Weesit Lee. (2017) A survey on biofeedback and actuation in wireless body area networks (WBANs). *IEEE Rev Biomed Eng*, 10:162–173, DOI: 10.1109/RBME.2017.2738009.

Muhammad Shadi Hajar, M. Omar Al-Kadri, and Harsha Kumara Kalutarage (2021). A survey on wireless body area networks: architecture, security challenges and research opportunities. *Computers & Security*, 104:102211, DOI: 10.1016/j.cose.2021.102211.

Texas Instruments (2008). The RS-485 Design Guide,. Rev. D.

Renesas. ISL3259E RS-485/RS-232 *Transciever Datasheet*, 8 2017. Rev. 2.

ANSI/TIA/EIA (2001). TIA/EIA-568-B.2: *Commercial Building Telecommunications Cabling Standard*. Rev. B.2

M. Patel and J. Wang, (2010) "Applications, challenges, and prospective in emerging body area networking technologies," in IEEE Wireless Communications, vol. 17, no. 1, pp. 80-88, February 2010, doi: 10.1109/MWC.2010.5416354

Li, HB., Kohno, R. (2008). Body Area Network and Its Standardization at IEEE 802.15.BAN. In: István, F., Bitó, J., Bakki, P. (eds) Advances in Mobile and Wireless Communications. Lecture Notes Electrical Engineering, vol 16. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-79041-9_12