SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 14, No. 1, February 2012, 23-33

UDC: 004.7:621.398]:621.317 DOI: 10.2298/SJEE1701023N

Multisensory Platform Based on NEC Protocol

Đorđe Novaković[1], Platon Sovilj[1], Nemanja Gazivoda[1]

Abstract: Multisensory signal acquisition represents one of the main concepts necessary to perform measurements in various industrial and consumer-oriented applications. This paper presents a development platform which provides a data acquisition from multiple sensors. The main module of the platform is the UNIDS-3 development board with PIC18F8520 microcontroller. Data acquisition from the sensors is performed on user demand by remote control. NEC protocol is implemented and IR receiver TSOP31238 is used. The measurement data are sent to a computer, which performs digital data processing, data visualization and data storage.

Keywords: Multisensory platform, Remote control, NEC protocol.

1 Introduction

Modern measurement systems are often based on transducing physical quantity to electrical signal by sensors [1 – 3]. Multisensory signal acquisition represents one of the main concepts that is necessary to perform measurements in various industrial and consumer-oriented applications [4, 5]. This paper presents a multisensory development platform based on NEC protocol [6, 7]. The presented platform is intended for design and development of smart home and IoT solutions [8 – 10], in which the solution parameters and functioning can be remotely controlled.

The control signal transmitter is a standard IR-based remote control device, and the receiver module is implemented by TSOP31238. After processing the control request, the microcontroller performs data acquisition of the platform sensors and sends the acquired data to the computer by UART communication. The computer processes the received data and performs the primary data processing, data visualization and storages the data into the hard disk memory. It should be noted that in this processes it is very important to provide the appropriate synchronization of all the platform modules.

The following measurement modules are included in the platform: temperature sensor DS18B20, linear accelerometer, four-channel 12-bit AD converter MCP3204 (intended for extension of the platform with analog

[1]Faculty of Technical Sciences in Novi Sad, University of Novi Sad, Trg Dositeja Obradovića 6, 21000, Novi Sad, Serbia; E-mails: n.djole.93@hotmail.com; platon@uns.ac.rs, nemanjagazivoda@uns.ac.rs

sensors) integrated on UNIDS-3 board, microphone module and motion sensor. It is important to emphasize that this platform can be also extended with various digital sensors.

Comparing this platform to popular commercial platforms for development of Smart Home and IoT applications, like ThingWorx IoT Platform, IBM Watson, Cisco IoT Cloud Connect, Salesforce IoT Cloud Carriots and General Electric's Predix [11], it is obvious that commercial platforms have more possibilities for supporting various communication protocols. However, the platform presented in this paper is simpler yet powerful, thus being more suitable for introduction of the students to the practical knowledge of Smart Home and IoT applications. This platform enables faster learning for students and faster design and development of educational student projects, which also can be simply extended to commercial projects.

2 Measurement System

The main element of the measurement system is PIC18F8520 microcontroller placed on UNIDS-3 development system, which contains a large number of integrated modules and peripherals, thus being suitable for design and development projects. PIC18F8520 is a 8-bit microcontroller which supports a large number of communication protocols, which makes it convenient for communication with a wide variety of peripherals. Some of the most important characteristics of this microcontroller are: two UART channels, SPI, I^2C, five timers and 16-channel AD converter. UNIDS-3 board represents a development system for the development of microcontroller firmware. UNIDS-3 system.

Microcontroller firmware is written using the MikroC integrated development environment (IDE), which is based on C programming language. This IDE was selected because it is characterized by simplicity of code writing, as well as a large number of already existing libraries which enable fast development of the firmware.

As one of the sensing and measurement elements, an accelerometer sensor is implemented, integrated on the module called “Three - Axis Accelerometer Board”. Operation of this module is based on the microcontroller integrated 10-bit AD converter, so it is necessary to connect the X, Y and Z outputs of the accelerometer sensor onto the appropriate analog inputs of the microcontroller.

The second sensor is a digital temperature sensor DS18B20 which has 12-bit resolution, and is characterized by the ability to measure temperature in the range from –55°C to +125°C. The useful characteristic of this sensor is that it uses OneWire communication protocol, so it can be connected to the microcontroller via a single pin.

The third sensing element is a microphone integrated into the module ”Microphone AMP board” with the appropriate amplification circuit. The amplification of this module can be adjusted by a potentiometer which is also integrated in the module. Measurement of this module is also performed by AD converter integrated into the microcontroller.

The fourth sensor represents a motion detector, integrated into the module ”Motion sensor board”, where a passive infrared light indicator AMN11112 represents a sensing component. This module generates a logic one when movement occurs or the logic zero when there is no movement. Readings from this sensor are done by reading the logical state on microcontroller digital input pin.

The last implemented element is the infrared receiver TSOP31238, used to receive signals from the remote control based on NEC protocol. This sensing element is characterized by very low power consumption, as well as internal filter that makes it resistant to interference. It is important to note that the carrier frequency of the remote control and infrared receiver must be the same, to ensure proper request decoding. In this platform this frequency is 38 kHz. Used remote control device has 21 buttons, enabling generation of 21 different requests. The remote control device and its defined button codes is shown in Fig. 1.

Fig. 1 – *Remote control device with defined button codes (left) and image of remote control (right).*

Block diagram of the measurement system, as well as all connections and directions of communication flow between them, is presented in Fig. 2. The connection between the remote control device and the IR receiver is represented with dotted line in the picture. The dashed rectangle frames the elements implemented in the UNIDS-3 board.

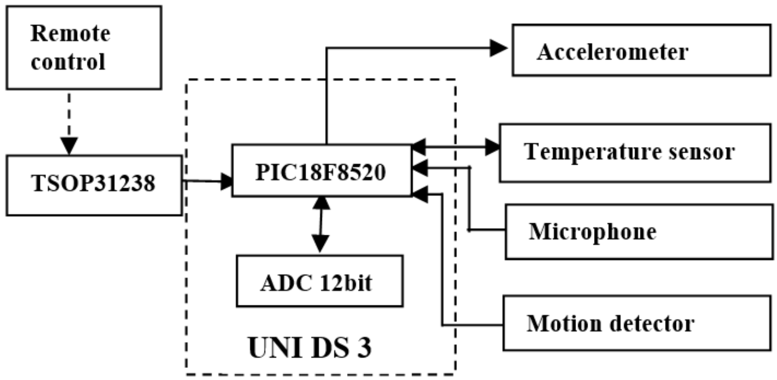

Fig. 2 – *Block diagram of measurement system.*

3 Firmware Implementation

The firmware of the microcontroller can be divided into several different segments, two of which (main program and interrupt routines) will be explained in following subsections.

3.1 Interrupt routine

Interrupt routine is the most important part of the firmware, responsible for receiving control signal packages based on NEC protocol. TSOP31238 is connected directly to external microcontroller interrupt pin, and the rising edge of the signal, at the microcontroller interrupt pin, generates the interrupt. It is followed by execution of the interrupt routine which performs control package processing and interpretation. The algorithm of the interrupt routine is shown in Fig. 3.

After the generation of the first interrupt, time measurement is started. Time measurement is implemented by using a 16-bit microcontroller timer TMR1. At the beginning of each generated interrupt, the timer register is read out. The difference between the old state of the timer register and the new reading is calculated as the elapsed time, which is in proportion with the frequency of the oscillator. If it turns out that the elapsed time between the generation of two interrupts is 13.5 ms, then this sequence can be decoded as the start signal, which marks the reception of the entire package.

Upon receipt of starter package, follows the determination of NEC logical one or NEC logical zero, depending on the time elapsed. For NEC logical one the elapsed time is 2.25 ms, for logical zero NEC elapsed time is 1.125 ms. When logic NEC level is decoded, it is entered into appropriate position of a character string variable intended for storing decoded values. At the same time, the number of decoded logical NEC levels is tracked, and when it reaches 32, i.e. when 32 bits of data are received, the reconstruction of the request is finished.

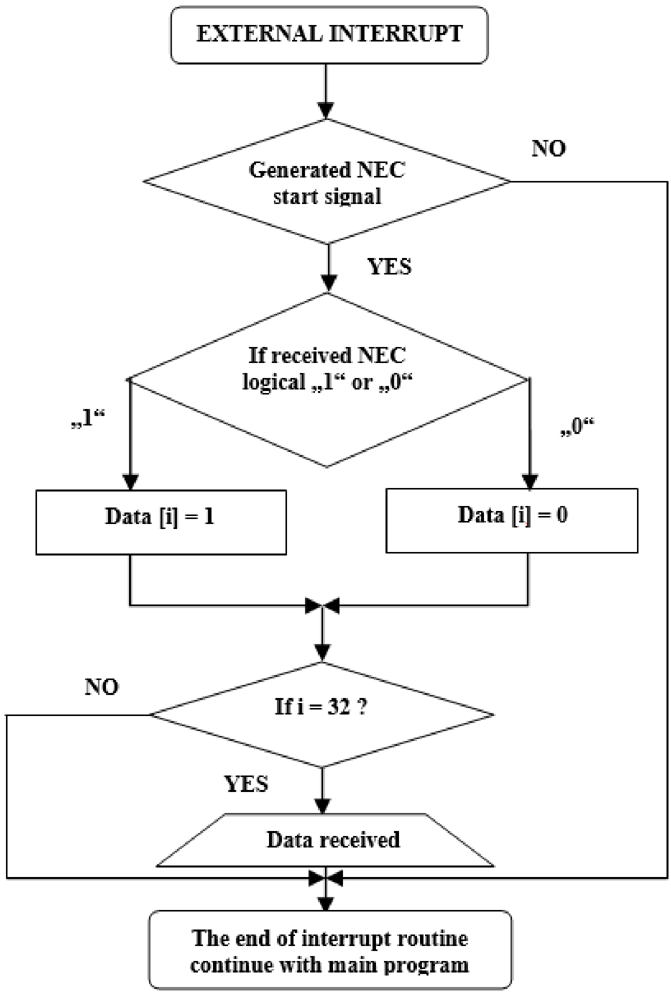

Fig. 3 – *Algorithm of the interrupt routine.*

3.2 Main routine

The main routine, after request interpretation, executes the targeted sensors code and sends the measurement data to the computer in the form of prepared data package. The algorithm of main routine is shown in Fig. 4.

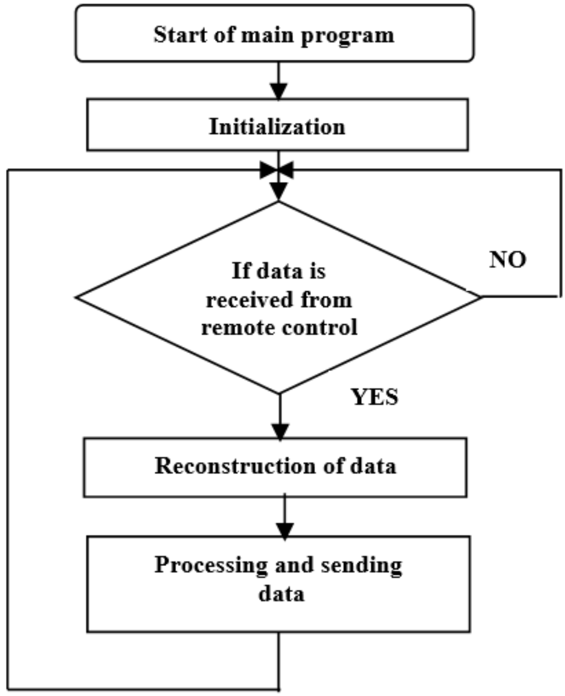

Fig. 4 – *Algorithm of the firmware routine.*

The main firmware starts with initialization commands which include:

- adjusting the microcontroller ports as input or output, i.e. as analog or digital;
- configuration of TMR1 module;
- enabling generation of external interrupt on the rising edge of the signal;
- SPI and I2C communication protocols initialization;
- enabling OneWire communication;
- setting UART communication at data transfer rate of 9600 bps.

After initialization, an infinite loop is implemented, by which the program waits for generation of control request from the remote control device. When the interrupt routine performs decoding of a signal as shown in Fig. 4, reconstruction of data is completed. The reconstruction is implemented as a function that accepts a character string of eight ones or zeros, and returns reconstructed integer value. With this function, address value, inverted address, data and inverted data are reconstructed, providing necessary data values. After the reconstruction of data, measurement and data acquisition by the appropriate sensor is performed, and then the acquired data are sent via the UART communication to the computer for further processing. When the generated task is performed, the program waits for a generation of new request. It should be noted that in addition to waiting for the request, the program acquires data from the motion detector, and continuously sends the information about movement activities in the sensed environment, to the computer.

4 Software Implementation

Computer software is developed in the C# programming language. The program is executed in two software threads: one thread is responsible for receiving data via UART communication, while the other thread is dedicated to data processing, data visualization and data storage activities. [16, 17] Although the program is being executed in two synchronized threads, for easier understanding, the software algorithm is shown in Fig. 5.

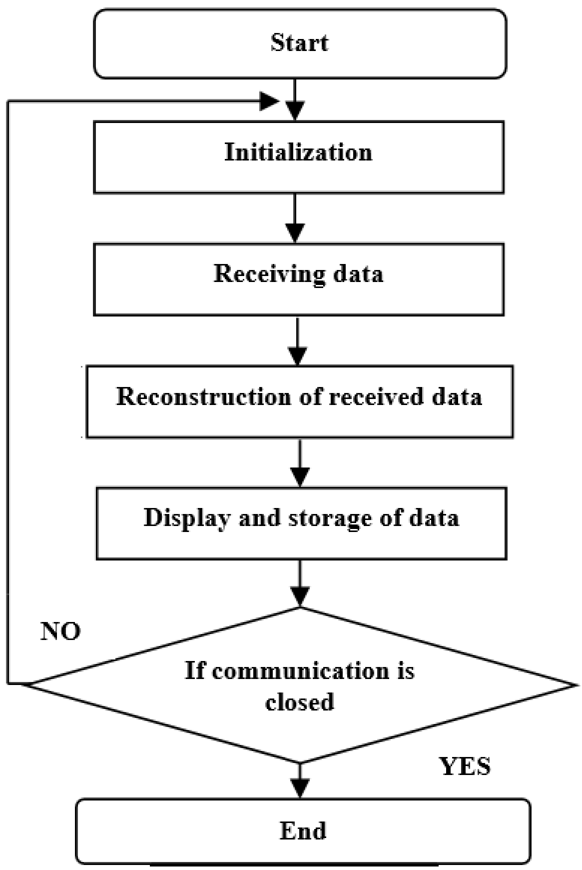

Fig. 5 – *Algorithm of the software.*

The program begins with initialization of communication where the user can define the computer port dedicated to the communication with microcontroller system. The transmission data rate is fixed to 9600 bps (the same rate as at the microcontroller side), in order to avoid user caused errors by manual configuration of data rate. In addition to initialization of communication, creation of program thread for data receiving is being performed.

After establishing the communication, receiving of data is being executed. Data are organized into packages where the first letter in the package indicates from which sensor the data packet originates. When the data are received, the next step is reconstruction of the data followed by visualization of reconstructed values for the appropriate sensors. At the same time, the reconstructed values from the sensor are stored in the file system.

If the user stops the communication by user interface, the software shuts down the data receiving. The application GUI form, which displays current sensors data is shown in Fig. 6.

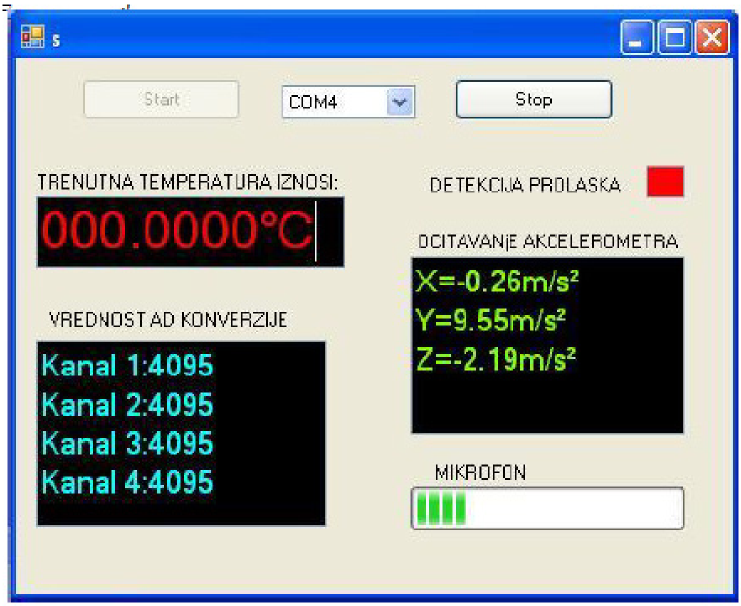

Fig. 6 – *The application GUI form.*

The system hardware which is consisted of UNIDS-3 board with microcontroller PIC18F8520, and all peripherals (control device and sensing modules) is shown in Fig. 7.

In **Table 1**, **Table 2** and **Table 3**, sensors data measured during a test session are presented.

Table 1
Data samples of acceleration.

X [m/s^2]	Y [m/s^2]	Z [m/s^2]
1.81	-0.9	-9.42
1.68	-1.16	-9.42
1.81	-1.16	-9.42

Table 2
Data samples from channels of ADC.

Channel 1	Channel 2	Channel 3	Channel 4
4095	4095	4095	4095
4095	4095	4095	4095

Table 3
Data samples of temperature, detection of movement and microphone.

Temperature [ºC]	Detection of movement	Microphone samples
021.2500	0	0001
021.2500	0	0047
022.0625	1	0127
022.1250	1	0082

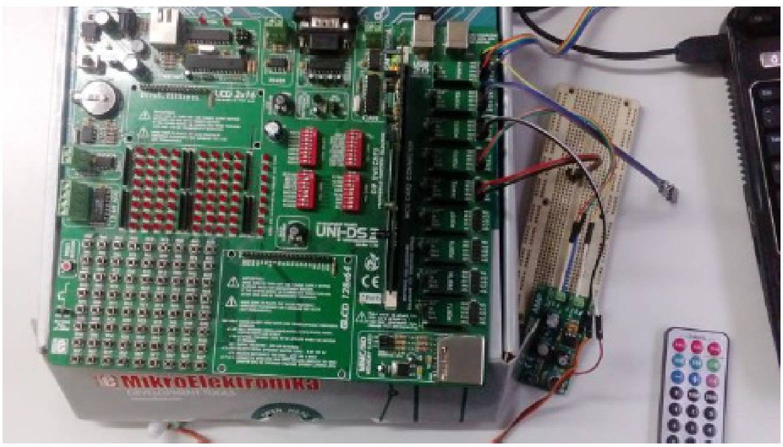

Fig. 7 – *The entire system hardware.*

5 Conclusion

This paper presents a platform intended for design and development of smart home and IoT solutions, in which the solution parameters and functioning can be remotely controlled. This platform achieved the goal of developing a platform which enables fast learning for students and fast design and development of educational student projects.

The most important design and development step to follow is the integration of microcontroller on a printed circuit board with the necessary additional modules, to achieve optimal dimensions of the platform. The next design step is replacement the current remote communication subsystem with a Bluetooth communication module, which would make the device more portable. Another benefit regarding this design improvement is the possibility to connect the system with smart mobile phones or tablets, and to create an Android application, which further opens a wide spectrum of possibilities for improvement of the user controllability of the system. Finally, regarding software application improvement, the improvement could be also the implementation of the connection to the cloud storage system, appropriate for storing the measurement data in longer periods of time. Besides enlarging the storage capacities of the system, this would also open the possibility for using the measured data by various remote devices.

6 Acknowledgement

This work was supported in part by the Provincial Secretariat for Science and Technological Development of Autonomous Province of Vojvodina (Republic of Serbia) under research grant No. 114-451-2800-2016-02 and supported in part by the Ministry of Education, Science and Technological Development of the Republic of Serbia under research grant No. TR32019.

7 Literature

[1] J. Fraden: Handbook of Modern Sensors: Physics, Designs and Applications, Springer, NY, USA, 2004.

[2] M. Popovic: Sensors and Measurements, Institute of Textbooks and Teaching Aids, East Sarajevo, Republik of Srpska, BIH, 2004. (In Serbian).

[3] Practical Design Techniques for Sensor Signal Conditioning, Analog Devices, 1999. Available at: http://www.analog.com/en/education/education-library/practical-design-techniques-sensor-signal-conditioning.html.

[4] R. Viswanathan, P.K. Varshney: Distributed Detection with Multiple Sensors: Part I – Fundamentals, Proceedings of the IEEE, Vol. 85, No. 1, Jan. 1997, pp. 54 – 63.

[5] R.S. Blum, S.A. Kassam, H.V. Poor: Distributed Detection with Multiple Sensors: Part II – Advanced Topics, Proceedings of the IEEE, Vol. 85, No. 1, Jan. 1997, pp. 64 – 79.

[6] R. Stair, G. Reynolds: Fundamentals of Information Systems, Cengage Learning, Stamford, CT, USA, 2013.

[7] NEC Infrared Transmission Protocol. Altium, 2013. Available at: http://techdocs.altium.com/display/FPGA/NEC+Infrared+Transmission+Protocol.

[8] T. Yamazaki: The Ubiquitous Home, International Journal of Smart Home, Vol. 1, No. 1, Jan. 2007, pp. 17 – 22.

[9] M.H. Jin, C.J. Fu, C.H. Yu, H.R. Lai, M.W. Feng: I3BM ZigBee Positioning Method for Smart Home Applications, International Journal of Smart Home, Vol. 2, No. 2, April 2008, pp. 127 – 139.

[10] L. Atzori, A. Iera, G. Morabito: The Internet of Things: A Survey, Computer Networks, Vol. 54, No. 15, Oct. 2010, pp. 2787 – 2805.

[11] A. McEwen, H. Cassimally: Designing the Internet of Things, John Wiley and Sons, London, UK, 2013.

[12] J. Sharp: Microsoft Visual C# 2008: Step by Step, Microsoft Press, Redmond, WA, USA, 2008.

[13] V. Milanovic: Interface Programming in C#, Infoelektronika, Nis, Serbia, 2011. (In Serbian).

[14] PIC18F8520 Microcontroller Datasheet, Microchip Technology, 2004. Available at: http://ww1.microchip.com/downloads/en/DeviceDoc/39609b.pdf.

[15] DFU IR Protocol (NEC) - IR Remote Controller, Ad Notam, 2013. Available at: https://www.ad-notam.com/download/RS232/ad_notam_IR_protocol_DFU.pdf.

[16] UNI-DS3 Development System. MikroElektronika, 2010. Available at: http://www.mikroe.com/downloads/get/874/unids3_manual_v103.pdf.

[17] TSOP31238 - IR Receiver Modules for Remote Control Systems Datasheet, Vishay, 2016. Available at: http://www.vishay.com/docs/82492/tsop312.pdf.