Metrol. Meas. Syst., Vol. XX (2013), No. 4, pp. 565–580.

PAN
POLSKA AKADEMIA NAUK

METROLOGY AND MEASUREMENT SYSTEMS

Index 330930, ISSN 0860-8229
www.metrology.pg.gda.pl

M&MS

DESIGN AND IMPLEMENTATION OF PLUG-AND-PLAY ANALOG RESISTANCE TEMPERATURE SENSOR

Nenad Jevtic[1], Vujo Drndarevic[2]

1) University of Belgrade, Faculty of Transport and Traffic Engineering, Belgrade, Serbia (✉ n.jevtic@sf.bg.ac.rs, +381 64 3527775)
2) University of Belgrade, School of Electrical Engineering, Belgrade, Serbia (vujo@etf.rs)

Abstract

In this paper the design and implementation of a plug-and-play analog resistance temperature sensor is presented. The smart temperature sensor consists of an analog sensor element with transducer electronic data sheet (TEDS) memory device and a network- capable application processor (NCAP) connected through a mixed-mode interface (MMI). The mixed-mode interface and NCAP front-end electronic support have been implemented by the use of a standard 8-bit microcontroller. NCAP's application processing and network communication functions are implemented based on the concept of virtual instrumentation using a PC. The implemented NCAP can also be used as a plug-and-play stand-alone data acquisition system or as development system for plug-and-play sensors compliant with the IEEE 1451.4 standard. Details of sensor implementation and test results are included in the paper.

Keywords: smart sensor, PT100, IEEE 1451.4, mixed mode interface, TEDS, NCAP, LabView.

1. Introduction

A modern measurement system should meet growing demands for measuring the increasing number of different physical quantities using a large number of sensors often distributed over a wide area. On the other hand, configuration and maintenance of various sensor types, especially when it comes to complex measurement systems, is a demanding and expensive task [1]. Experience shows that about 20% of the total cost of a data acquisition system belongs to the hardware configuration and calibration of sensors [2]. In addition, problems related to common errors that occur during the manual entry of important parameters of the sensor, such as measuring range, sensitivity, scale factors etc., should be also considered. Solution to this problem appeared in the form of smart sensors that enable the plug-and-play concept and provide automation of the process of sensor connection, configuration and maintenance.

Recognizing the numerous benefits offered by the plug-and-play concept, application of this technology in connecting analog sensors has been the subject of interest in a number of papers [3-5]. A novel system for distributed measurement of gamma radiation has been described in [3]. The system is composed of multiple intelligent sensor nodes connected through Ethernet to a client PC that executes application software. A microcontroller-based minimal 1451.4 NCAP is presented in [4]. The proposed NCAP enables read and write TEDS functions with a PC via the RS-232 serial link. A solution for embedding smart sensors in a wireless air quality monitoring network consisting of minimal 1451.4 NCAP and serial to wireless bridge is discussed in [5].

In parallel with the research work relating to plug-and-play sensor configuration, numerous manufacturers of sensors and measuring systems offered their solutions for the sensor interfacing and configuration problem, which have not become generally accepted [6].

Article history: received on Feb. 11, 2013; accepted on Aug. 01, 2013; available online on Dec. 10, 2013; DOI: 10.2478/mms-2013-0048.

To promote widespread adoption of plug-and-play sensors, the National Institute of Standards and Technology (NIST) together with the Institute of Electrical and Electronics Engineers developed IEEE P1451.4 standard for self-describing analog sensors using standardized Transducer Electronic Data Sheets (TEDS) [7, 8]. The Transducer Electronic Data Sheet, proposed in IEEE 1451.4, uses a very compact form to store sensor data. Standard templates for encoding sensor information are provided for most frequently used sensors, but several papers [9-11] state that these templates are not suitable for some types of sensors. A method for storing gas sensor information by the use of standard templates is described in [9]. Also, a mixed-mode interface able to automatically configure itself on the basis of TEDS information is implemented. A method of storing information for electronic tongue sensors is discussed in [10] and a new standard template for the electronic tongue system is proposed. An effort to apply the IEEE 1451.4 concept to frequency-output sensors is described in [11].

Industrial smart modules with PT100 inputs and various communication interfaces are readily available on the market place, but very few of the proposed solutions offer plug-and-play connectivity to the data acquisition system or to the network [12, 13]. This paper focuses on the process of design and implementation of a plug-and-play resistance temperature detector based on standard industrial PT100 sensor. To support realization of the plug-and-play sensor, appropriate hardware and software tools are developed.

In subsequent sections, different components of the system are detailed. In section 2 the process of implementation of a mixed-mode interface for RTD sensors is given. For interfacing of an RTD sensor to the arbitrary network, a microcontroller based circuit is used and its design details are presented in section 3. Development of tools necessary for generating and editing of a Transducer Electronic Data Sheet for plug-and-play sensors has been presented in section 4. A summary of the results and other possible applications of the proposed system are described in the final section.

2. Implementation of mixed-mode interface

To accommodate both a measurement signal and a serial digital signal for transferring TEDS data, the plug-and-play analog sensor includes a mixed-mode interface. In accordance with the IEEE1451.4 standard, two classes of mixed-mode interface are defined. With the Class 1 MMI interface for transmission of digital data and analog signals, the same pair of wires is being used, provided that the transfer of the transducer signal and digital TEDS data is performed sequentially. This way, the number of lines used for connection is reduced to a minimum. With MMI interface of Class 2, transmission of analog signals and digital communication is done through separate lines. Since lines for the transmission of analog signals and digital data are separated, with Class 2 of MMI interface transmission can take place continuously and simultaneously.

The temperature sensor consists of a traditional analog RTD sensor element and TEDS memory. Interfacing of a PT100 temperature sensor and TEDS memory device with the data acquisition system was done in accordance with Class 2 MMI interface. The PT100 sensor is attached to the front-end of the acquisition system using a four-wire connection as shown in Fig. 1. There are many convenient memory chips on the market suitable for TEDS data storage. In this application, Maxim/Dallas EEPROM DS2433 [14] with a capacity of 4096 bits was chosen for storing TEDS data. Digital communication over MMI Class 2 interface with the DS2433 EEPROM is realized in accordance with the IEEE 1451.4 standard by using a Data Transmission Protocol [7] based on Maxim/Dallas One-wire protocol [15].

One-wire interface (OWI) uses a simple protocol for serial communication that takes place on the master-slave principle. Master provides power through a common line and initiates all transactions with the slave according to strictly defined time sequences. Data and commands are read and written sequentially bit by bit, least significant bit first.

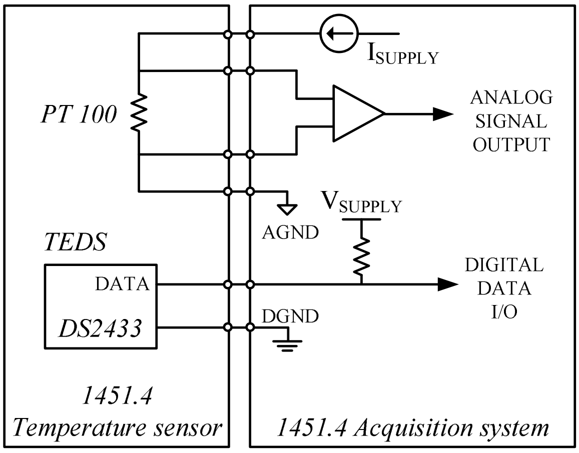


Fig. 1. PT100 sensor and TEDS memory interfacing.

Signaling on the One-wire bus is divided into fixed time slots. Time slot should last from 60 µs to 120 µs. Master starts every transmission by pulling the bus low and only one bit is sent or received during each time slot. There are three basic signals for sending and receiving bits: Write 1 – used to send logic 1 to device, Write 0 – used to send logic 0 to device, and Read – used to read bit from device. Write 1 time slot is initiated by pulling the bus low for 1 µs to 15 µs, and then releasing the bus until the end of time slot. Write 0 begins the same way, and then holding the bus low until the end of the time slot. Read time slot master initiates by pulling the bus low for 1 µs to 15 µs, and then releasing the bus, to either be pulled low or high by the slave device. The device should pull the bus low or high to send appropriate data bit to the master. Fig. 2a shows timing sequences for the Write and Read time slots.

There is one more synchronization signal Reset – used to reset all devices connected to One-wire bus. Reset signal initiates master by pulling the bus low for at least 480 µs, and then releasing the bus. After resetting, every device should respond with a Presence signal. The Presence signal should begin 15 µs to 60 µs after Reset and should last from 60 µs to 240 µs. The effect of the Reset signal is shown in the time diagram in Fig. 2b.

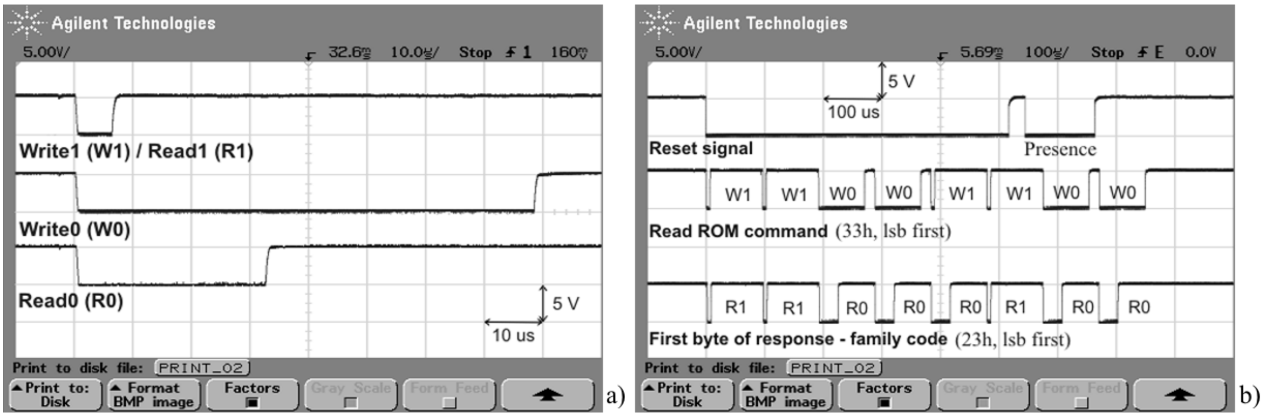


Fig. 2. Signaling over One-wire interface: Basic Read/Write signals (a),
and Reset, Read ROM command and device response signals (b).

Communication over the One-wire interface follows tree basic steps: device reset, selecting device, and send/receive data. Since more then one device can be present on the One-wire bus, communication starts with Reset signal issued by master. All devices on the bus respond with the Presence pulse. Presence pulse allows the master to determine that one or more transducers are present on the bus.

The second phase in communication sequence involves selecting a slave device for subsequent communications. Device selection is based on a 64-bit Unique Registration Number (URN) with a family code, serial number, and CRC field. For subsequent communication master can select all slaves, a slave with a specific URN number, or use a search algorithm to discover an unknown slave device. Since the URN number is permanently stored in device's ROM memory, commands for selecting target slave device are referred to as ROM commands. There are four ROM function commands: Search ROM, Match ROM, Read ROM and Skip ROM. Sending a Read ROM command is illustrated in Fig. 2b. As a result of receiving this command, the slave device sends a 64 bit URN stored in ROM memory. Fig. 2b shows the first byte of response of the selected device. Once a device with a specific URN has been selected, the master can issue commands to read or write EEPROM data.

Having successfully executed a ROM command, the slave device enters into the third phase of communication sequence – the memory function command state. The master then issues the appropriate command to read or write TEDS data to EEPROM memory of the selected device. After sending command for reading or writing data, the master sends one- or two-byte-long starting address, which depends on EEPROM memory type, and begins reading or writing data. When the read or write operation is finished, master must issue Reset signal to start the three-step communication sequence again.

3. Implementation of network capable application processor

To provide mixed-mode interface communication, communication between the RTD sensor and an arbitrary network and data conversation function, a Network Capable Application Processor (NCAP) was designed and implemented. The NCAP consists of a microcontroller-based front-end electronics and a PC. In addition to the functions listed, NCAP provides editing of TEDS data based on the concept of virtual instrumentation. Possessing these functions, the implemented NCAP can also be used as a stand alone plug-and-play Data Acquisition and Development System for 1451.4 sensors. Development of NCAP will be discussed in detail.

3.1. NCAP Hardware

NCAP hardware consists of a microcontroller-based interfacing circuit and personal computer (Fig. 3). The standard 8-bit Atmel AT90USB1286 [16] microcontroller is used for One-wire digital communication with TEDS memory, to program instrumentation amplifier gain, for USB communication with PC and for analog-to-digital conversion (ADC) of input signals. One-wire TEDS communication has been provided through serial UART interface embedded in the microcontroller.

To reduce any error because of the wire resistance, a platinum PT100 sensor is connected with the front-end of NCAP Data acquisition system using a four-wire interface. One pair of wires is used to connect the 1 mA excitation current source, while the other pair is used to connect the measuring voltage signal. The four-wire measurement ensures that the voltage drop across the wire resistance does not affect the PT100 voltage measurement due to zero current flow into the high input impedance of the instrumentation amplifier.

Electronics for signal conditioning consists of a programmable gain amplifier implemented with INA122 instrumentation amplifier and MCP41100 programmable resistor, and antialiasing Chebyshev second order low-pass filter with 10 Hz cutoff frequency.

The PT100 resistance is given by

$$R_{PT100} = \frac{V_{PT100}}{I_0} = \frac{V_{ADC0}}{G_{AMP} I_0}\,. \tag{1}$$

From (1) one can find the PT100 resistance, the current source I_0 and the amplifier-ADC path measuring the voltage must be accurate, which means free from offset, gain, and nonlinearity errors. To reduce the measurement error caused by the current source, instrumentation amplifier and ADC, a reference resistor was used. A constant current I_0 flows through a known very accurate reference resistor R_r connected in series with the PT100

$$I_0 = \frac{V_{ADC1}}{R_r}\,, \tag{2}$$

where the voltage across the reference resistor is measured by ADC1 and, according to (1), the PT100 resistance is given by

$$R_{PT100} = \frac{V_{ADC0}}{G_{AMP} V_{ADC1}} R_r\,. \tag{3}$$

As can be seen from (3), the accuracy of measurement of R_{PT100} does not depend on the accuracy and stability of the constant current source I_0.

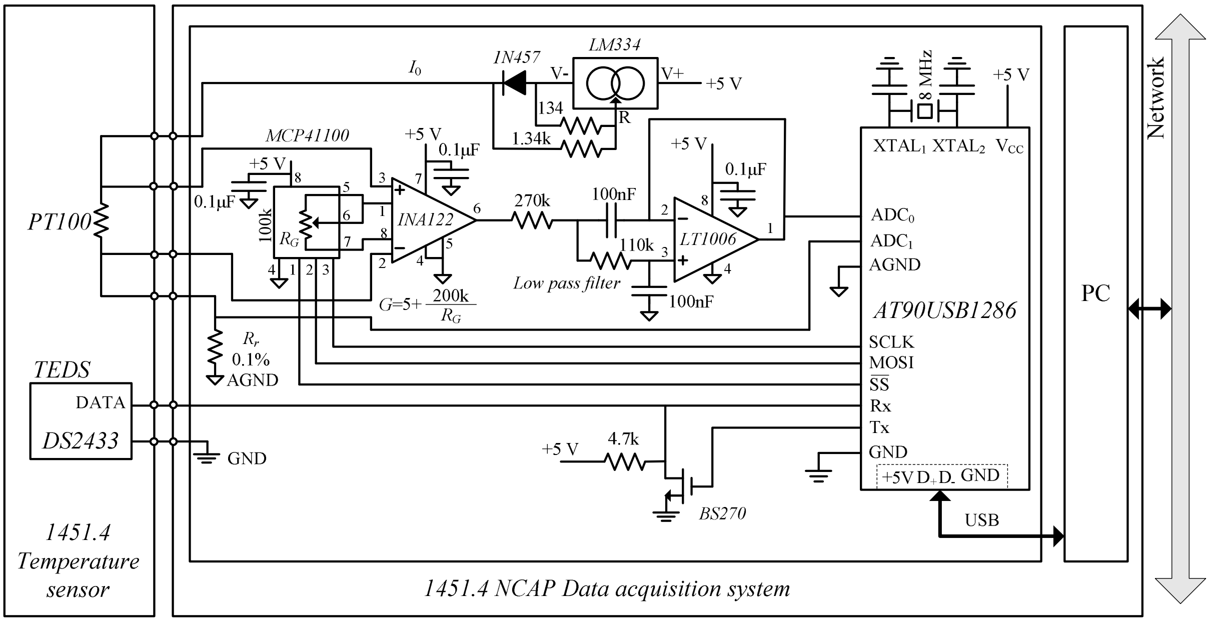


Fig. 3. Architecture of 1451.4 NCAP/Data Acquisition System.

If we assume that the ADC circuit introduces an offset, then the measured PT100 resistance, according to (3), is given by the equation

$$R_{PT100} = \frac{V_{ADC0} - V_{offADC0}}{G_{AMP}(V_{ADC1} - V_{offADC1})} R_r\,. \tag{4}$$

Offset for both ADC's channels, $V_{offADC0}$ and $V_{offADC1}$ are measured every second by internally shorting the ADC's inputs.

Signal conditioning circuits, together with the constant current source are optimized for use with resistance temperature detectors (RTD) but can be used for interfacing other types of analog sensors such as: resistance sensors, bridge sensors, strain gauges and thermistor sensors [17]. Analog-to-digital conversion of input signals are performed using a 10-bit successive approximation ADC embedded into the microcontroller. Both measurement data and TEDS data are sent to the PC computer over the USB interface.

A PC has been used for the software development and testing and to support NCAP functionality. The PC is also responsible for securing the network communication function. The PC, which can be in the form of a single board computer, was chosen as a part of the NCAP because it already has interface drivers for most supported network protocols and it is highly integrated with modern T&M and DAQ systems.

As can be seen from the above analysis, measurement errors are introduced by both NCAP electronics and the RTD sensor. Bearing in mind that this work is dedicated to the development of electronics for a plug-and-play temperature sensor, attention will be paid to the determination of measurement uncertainty caused only by the NCAP hardware.

Since the PT100 resistance measurement is done indirectly according to (1), the measurement uncertainty of the signal conditioning and data acquisition electronics is given by the equation [18, 19]

$$u(R_{PT100}) = \sqrt{\left(\frac{\partial R_{PT100}}{\partial V_{PT100}}\right)^2 u^2(V_{PT100}) + \left(\frac{\partial R_{PT100}}{\partial I_0}\right)^2 u^2(I_0)} = \sqrt{c_1^2 u^2(V_{PT100}) + c_2^2 u^2(I_0)}, \tag{5}$$

where: $u(R_{PT100})$ - uncertainty of PT100 resistance measurement, $u^2(R_{PT100})$ - voltage measurement variance, and $u^2(I_0)$ - current measurement variance.

Given the $V_{PT100} = V_{ADC0}/G_{AMP}$, the voltage measurement variance $u^2(V_{PT100})$ is defined by the following dependence

$$u^2(V_{PT100}) = \left(\frac{\partial V_{PT100}}{\partial V_{ADC}}\right)^2 u^2(V_{ADC}) + \left(\frac{\partial V_{PT100}}{\partial G_{AMP}}\right)^2 u^2(G_{AMP}) = c_3^2 u^2(V_{ADC}) + c_4^2 u^2(G_{AMP}), \tag{6}$$

where: $u^2(V_{ADC})$ - variance of ADC voltage measurement, and $u^2(G_{AMP})$ - variance of amplifier gain. In accordance with (2), the variance of current measurement I_0 is given by

$$u^2(I_0) = \left(\frac{\partial I_0}{\partial V_{ADC}}\right)^2 u^2(V_{ADC}) + \left(\frac{\partial I_0}{\partial R_r}\right)^2 u^2(R_r) = c_5^2 u^2(V_{ADC}) + c_6^2 u^2(R_r), \tag{7}$$

where $u^2(R_r)$ is the variance of the resistance of the reference resistor.

A simplified estimation of measurement uncertainty of the PT100 resistance can be made if we assume that the variance of voltage measurements is equal to the ADC resolution. The values of the variance for the voltage gain and for the resistance of the reference resistor were taken from the manufacturer's datasheet (Table 1).

Table 1. Variance of voltage measurement, resistance of reference resistor, and amplifier gain taken from the manufacturer's datasheet.

Quantity symbol	Distribution of probability	k	Maximal variance
$u^2(V_{ADC})$	rectangular	$\sqrt{3}$	$3.1789 \cdot e^{-7}$ [V^2]
$u^2(R_r)$	normal	2	$2.5 \cdot e^{-7}$ [Ω^2]
$u^2(G_{AMP})$	rectangular	$\sqrt{3}$	$3.0 \cdot e^{-6}$ [(V/V)^2]

Table 2 shows the components of uncertainty and the estimated overall measurement uncertainty of the NCAP hardware for the worst case when R_{PT100} = 390 Ω.

Table 2. Uncertainty budget of resistance estimate for the R_{PT100}=390 Ω.

Sensitivity coefficients	c_1	$1\cdot e^{-3}$ [1/A]
	c_2	$3.9\cdot e^{-5}$[V/A^2]
	c_3	$7.8\cdot e^{-2}$ [V/V]
	c_4	$3.042\cdot e^{-2}$ [V]
	c_5	$1\cdot e^{-3}$ [1/Ω]
	c_6	$1\cdot e^{-6}$ [V/Ω^2]
Standard uncertainty of current estimate $u(I_0)$		$5.6382\cdot e^{-7}$ [A]
Standard uncertainty of voltage estimate $u(V_{PT100})$		$6.8631\cdot e^{-5}$ [V]
Standard uncertainty of RT100 resistance estimate $u(R_{PT100})$		**0.2340 [Ω]**
Extended uncertainty (95%) of RT100 resistance estimate $U(R_{PT100})$		**0.4607 [Ω]**

The determined standard measurement uncertainty of PT100 resistance is 0.2340 Ω, and the extended uncertainty of resistance measurement with the confidence level assumed at 95% is 0.4607 Ω. These uncertainties do not exceed 0.12% of the value of the measured resistance.

3.2. Organization and implementation of software

NCAP software is organized in two modules. One software module is running on the PC while the other part runs on the microcontroller. Communication between the PC and the microcontroller operates through the USB interface. Organization of microcontroller and PC program modules is presented in Fig. 4.

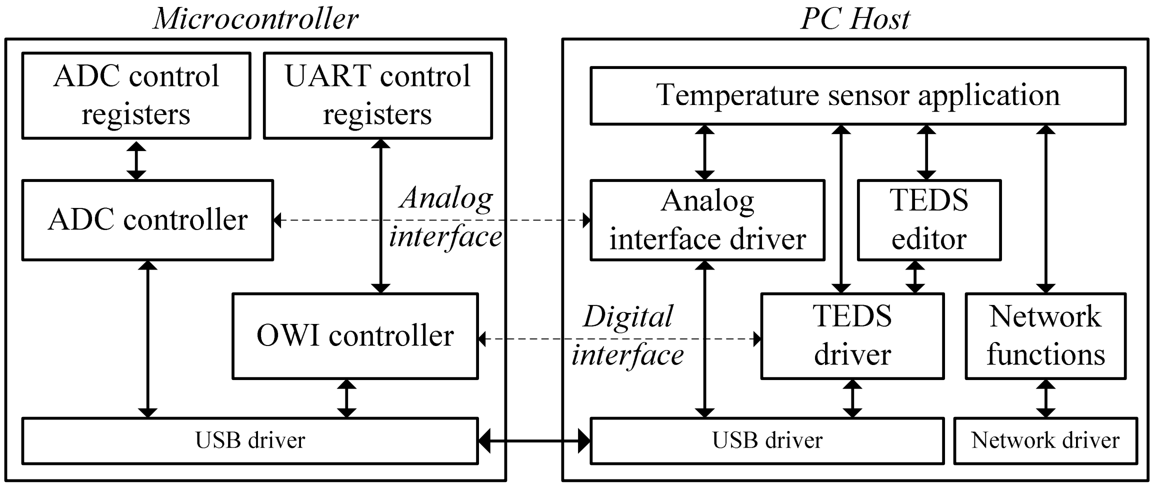


Fig. 4. NCAP software organization.

Microcontroller's software module (firmware) is organized in three main function blocks: *ADC controller*, *OWI controller* and *USB driver*. ADC controller block contains commands for configuration of front-end electronics and ADC, enabling and disabling ADC and reading samples from a specified channel. Communication with the TEDS memory device is performed using *OWI controller* block commands. Microcontroller's firmware *OWI controller* block contains commands for configuration of microcontroller UART to perform OWI interface signalling, reset One-wire devices, search for new slave devices, selecting device with specific URN, sending and receiving data. Microcontroller firmware is developed in C programming language using Atmel's AVR Studio [20].

The second part of the NCAP software refers to the application program running on the PC. PC application software performs, through the microcontroller firmware modules, downloading of TEDS data from the memory device, plug-and-play configuration of the PT100 sensor and display of the measurement results in engineering units. This program also provides network communication function. PC application software is developed under the concept of virtual instrumentation, using LabVIEW [21] development environment. USB driver functions are based on National Instrument's VISA application programming interface for USB bus.

Reading the contents of TEDS memory is performed after turning the power on, resetting of the system or on request by calling of corresponding *TEDS driver* functions. *TEDS driver* functions are performed by sending commands to microcontroller's firmware *OWI controller block*. After configuration of the transducer, during normal operation, 1451.4 transducer behaves as a classic analog transducer. Transmission of analog data is done by calling *Analog interface driver* functions. These functions are executed by sending commands to *ADC controller* block of the microcontroller firmware.

Both measurement data and digital TEDS data are transferred to the PC using USB driver functions. The USB bus allows simultaneous transmission of analog and digital data and control messages using three separate data links over the same bus.

The PC application program automatically searches for the memory device attached on One-wire bus using a search device algorithm implemented in microcontroller's firmware *OWI controller* block. The application program then chooses an appropriate TEDS driver component to provide communication with the discovered device. The implemented software supports DS2430A (256 bits), DS2431 (1024 bits), and DS2433 (4096 bits) EEPROM memory devices. Auto detection of a memory device is based on an 8-bit family code which is a part of unique registration number (URN) permanently stored in every One-wire device. After detecting a memory device, the application program then reads TEDS data and performs data parsing.

The process of parsing Transducer TEDS consists of several steps. After reading data stored in EEPROM device, the first step is to calculate and verify checksum bytes. Checksum bytes are used for error checking. For this purpose the data memory is split into pages (sections) of 256 bits (32 bytes). For every memory page an 8-bit checksum is made.

The next step in parsing TEDS data is to select the first 64 bits belonging to the Basic TEDS. Basic TEDS does not use standard data types such as integer, float or ASCII. Instead, numerical fields use the data type UnInt which is an unsigned integer number with variable length. For converting a Basic TEDS bit stream into data that the user can understand, an appropriate number of bits should be read for every field and then interpreted as a decimal number.

The Basic TEDS is separated from the next TEDS section by a two-bit field called Selector of descriptor (DS). If the Basic TEDS section is followed by an IEEE 1451.4 standard template, the value of the DS field is zero.

Standard template for the resistance temperature detectors begins with the template ID field, 8-bit integer number whose value is 37, followed by 16 property fields and two selectors. Each field is of different type and length. Most of these fields are easy to convert to decimal notation suitable to display, but some of property data types require special attention. These fields require appropriate interpretation defined in Standard [7]. For example, the measuring range of resistance temperature sensor PT100 is specified using two parameters: minimum and maximum temperature. These parameters are stored in TEDS memory as 11-bit *Constant resolution* (ConRes) data type. Constant resolution data type provides linear mapping of floating point data. The next RTD sensor property that needs attention is excitation current. Excitation current property is stored as 8-bit *Constant relative resolution* (ConRelRes) data type. Constant relative resolution data type provides logarithmic mapping of floating point data.

Standard transducer template TEDS section can be followed by Calibration template section. Transducer templates store data for particular transducer class and therefore provide the foundation for the automatic transducer detection and configuration process. Calibration templates take this process one step further by storing data for individual transducers. There are three calibration templates: Calibration table, Calibration curve, and Frequency response table.

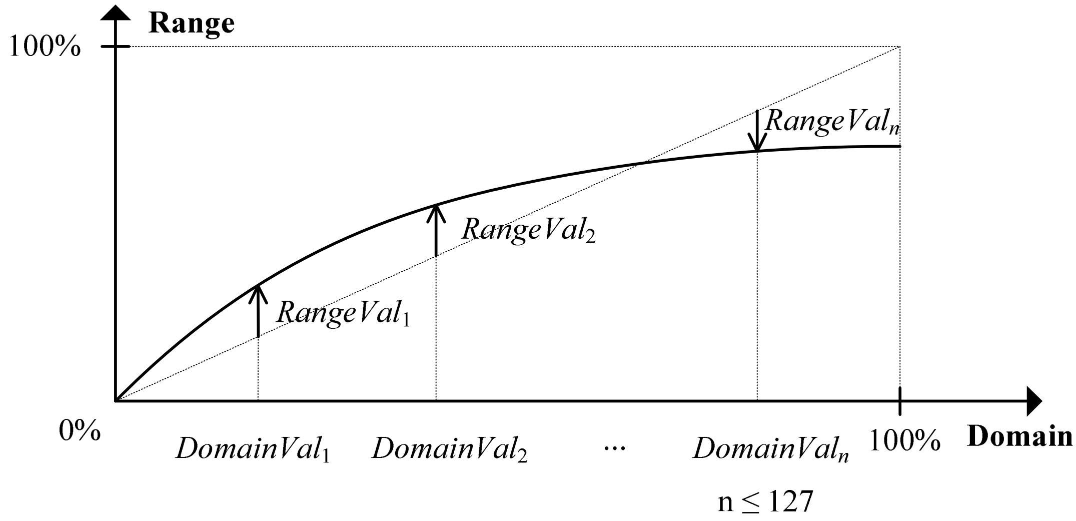


Fig. 5. Calibration table transfer function.

The *Calibration table* template describes a lookup table with the electrical-to-physical transfer function of a sensor. The Calibration table specifies deviations from an ideal transducer transfer function (Fig. 5). Every calibration point is specified with two parameters: DomainVal and RangeVal. DomainVal is expressed as a percentage of the full range of the sensor. RangeVal represents the deviation from the expected value, expressed as a percentage of the full range of the sensor.

The *Calibration curve* template specifies the transfer function of a transducer as a multi-segment polynomial curve. Each segment (up to 255 segments) is defined by start of segment parameter, number of polynomial coefficients for segment (up to 127), and array of polynomial coefficients and powers.

The *Frequency response* table template describes the amplitude-versus-frequency transfer function of a transducer in a similar manner as the calibration table template.

3.3. Interfacing PT100 temperature sensor

In accordance with the presented concept, the plug-and-play temperature measurement system is implemented and tested. This system consists of a PT100 resistance temperature detector, TEDS memory device, front-end electronics and microcontroller based board and PC computer (Fig. 6). The microcontroller based front-end electronics and PC form the NCAP, and they are interconnected through the USB bus. The temperature sensor is connected to the NCAP through the mixed-mode interface.

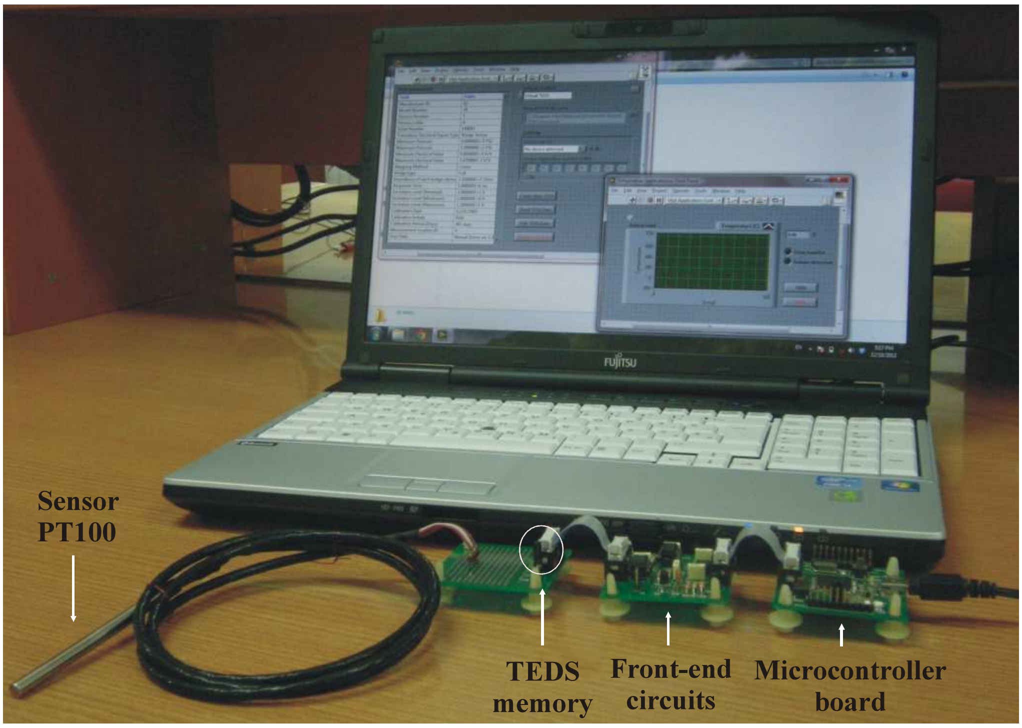


Fig. 6. Implemented plug-and-play temperature measurement system.

After power-on, reset or on request PC application software automatically configures the system through the following sequence:

- Search algorithm is launched to discover attached One-wire memory devices,
- TEDS data are read from the memory device,
- Parsing of TEDS data in accordance with standard template ID,
- Autoconfiguration of front-end electronics based on extracted sensor's information.

Upon successful completion of the configuration process, the application periodically reads analog data, performs data conversion and signal processing and displays measured results in appropriate physical units (Fig. 7).

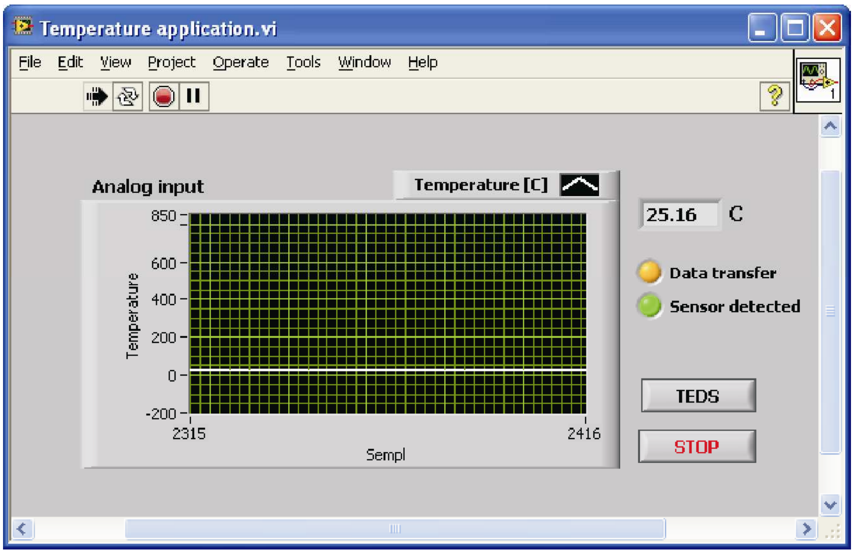


Fig. 7. Graphical User Interface for temperature measurement with plug-and-play RTD sensor.

To provide automatic sensor identification and automatic configuration, TEDS data are stored in a One-wire EEPROM device according to standard template 37. This template defines parameters such as temperature range, resistance range, excitation current, and other parameters relevant to resistance temperature detectors.

For the platinum RTD sensor PT100 used, the minimum physical value is -200 °C and the maximum physical value is +850 °C. In such a wide temperature range, the sensor transfer function shows considerable nonlinearity. This nonlinearity can be well approximated with two polynomial equations, referred to as Callendar–Van Dusen equations [7]:

$$R(t) = R_0\,[1 + At + Bt^2 + C(t–100°\text{C})t^3],\ (t < 0°\text{C}), \tag{8}$$

$$R(t) = R_0\,[1 + At + Bt^2],\ (t \geq 0°\text{C}), \tag{9}$$

where t is RTD sensor temperature in °C, $R(t)$ is resistance at temperature t, R_0 is resistance at 0 °C. The values of R_0, A, B, and C for PT100 are specified in IEC 60751 for standard industry-grade platinum and are: R_0 = 100 Ω, $A = 3.9083{*}10^{-3}$ °C^{-1}, $B = –5.775{*}10^{-7}$ °C^{-2}, $C = –4.183{*}10^{-12}$ °C^{-4}. The standard TEDS template, in addition to the listed values for A, B and C referred as *Case1*, contains five more sets of Callendar–Van Dusen coefficients for the best polynomial approximation of the transfer function of PT100 that is being in use. After the precise measurement of sensors resistance, the measured temperature is determined by (8) for negative, or by (9) for positive temperatures. For fully specifying the transfer function for the PT100 sensor, standard Calibration table template (ID=40) or Calibration curve template (ID=41) can be used in conjunction with the standard transducer template (ID=37).

Based on TEDS data for the excitation current and sensor's resistance range, a PC program automatically calculates the value of R_G resistance to adjust instrumentation amplifier gain. For 1 mA excitation current and maximum resistance 390.26 Ω, the amplifier gain should be G = 5 V / (1 mA * 390.26 Ω) = 12.81. This is achieved by adjusting R_G=25.6 kΩ (see Fig. 3).

To improve the accuracy of the system, compensation of front-end electronics is performed using a precision resistance, as explained previously. This procedure compensates for internal current source and ADC offset errors.

In the absence of an environmental test chamber and certified thermometer, testing of the realized plug-and-play sensor was carried out at 25 °C using a series of resistors of known resistance instead of using PT100 sensor. In order to verify the measurement accuracy, the obtained results are compared with measurements taken with an Agilent 34401A DMM [22]. The measurements were carried out for eight resistors in 1000 points. Table 3 shows the results of resistance measurements which were carried out in parallel, with DMM and with our NCAP Data acquisition system. The equivalent temperature is calculated using Callendar–Van Dusen equations (8)-(9).

Table 3. Comparative test results of resistance measurement using plug-and-play sensor and reference instrument.

Nominal value		**Agilent 34401A**			**PnP sensor**		
R_{nom} [Ω]	Tol. [%]	R_{meas} [Ω]	St. uncertainty [Ω]	Equi. temp. [°C]	R_{meas} [Ω]	St. uncertainty [Ω]	Equi. temp. [°C]
18	1	17.9611	8.79e-6	-201.29	17.9164	6.9e-4	-201.40
51	1	50.7311	1.495 e-5	-123.36	50.6701	1.5e-3	-123.51
100	1	99.8820	9.92 e-6	-0.30	99.6088	3.6e-3	-1.00
150	1	149.402	1.594 e-5	128.86	149.2097	5.6e-3	128.34
200	1	199.711	1.391 e-5	265.55	199.3970	6.0e-3	264.67
270	1	267.957	3.383 e-5	461.17	267.9073	7.6e-3	461.02
330	1	328.339	1.701 e-5	645.88	328.0192	9.1e-3	644.88
390	1	390.608	1.798 e-5	850.43	390.9000	1.14e-2	851.43

Test results from Table 3 show that the resistance measurement error originating from the NCAP electronics does not exceed ±0.32 Ω, which means that the resistance relative error does not exceed 0.27%, and that the temperature error does not exceed ±1°C over the entire measuring range. If more precise measurements are needed, an external high resolution AD converter can be used with embedded programmable gain amplifier.

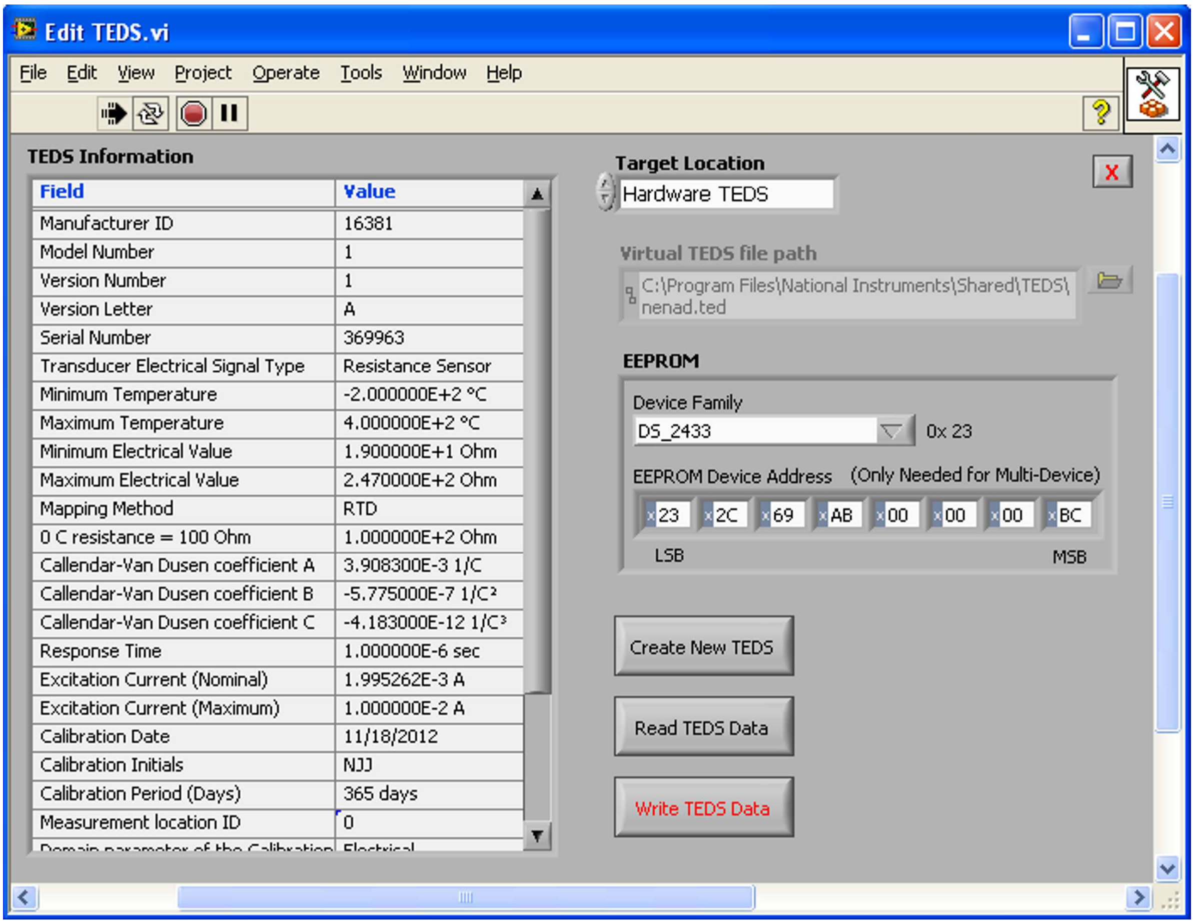
4. TEDS Editor

The essential part of the smart plug-and-play transducer is TEDS. TEDS contains basic information such as manufacturer ID, serial and model number, but also can contain data describing the physical and electrical characteristics of a transducer, calibration date and period, transducer location and other parameters. To facilitate the reading, writing and editing of TEDS data using standardized Templates, a PC-based TEDS editor is developed. The TEDS editor allows a user to create new TEDS data, read and edit existing TEDS and write data to virtual TEDS (VTEDS) file or an EEPROM memory device.

TheTEDS editor is developed in LabVIEW environment applying the concept of virtual instrumentation (VI). For the transfer of TEDS data between the PC application program and EEPROM memory, the already implemented TEDS driver VI library, described in the previous section, is used. Parsing of TEDS template is done using VI collection called TEDS library [23, 24], developed by National Instruments. This VI collection helps the application software developer to parse TEDS data acquired from the transducer or to read/write these data into the VTEDS file. Fig. 8 shows the front panel and Fig. 9 the block diagram of the TEDS editor.

Upon starting the TEDS editor, a user first selects whether to use the EEPROM memory or virtual TEDS file. By selecting *Hardware TEDS* target location (Fig. 8), the user can read or write TEDS data to the EEPROM memory. Except reading and writing TEDS data to EEPROM memory, the TEDS editor also provides read and write functions to a virtual TEDS file. By selecting *Virtual TEDS* target location, the user can read or write TEDS data to a file. If *Virtual TEDS* option is selected, the appropriate target file path should also be entered. A virtual TEDS file can be stored on a local computer for testing purposes, or when the memory device can not be embedded in the sensor.

Following the selection of the appropriate target location, the user runs one of the three sub-VIs: *Read TEDS Data*, *Create New TEDS*, and *Write TEDS Data* (Fig. 9). By selecting *Read TEDS Data* from TEDS editor GUI, the application starts sub-VI that searches for attached transducer's EEPROM device and reads TEDS data using the One-wire protocol.



Edit TEDS.vi

File Edit View Project Operate Tools Window Help

TEDS Information

Field	Value
Manufacturer ID	16381
Model Number	1
Version Number	1
Version Letter	A
Serial Number	369963
Transducer Electrical Signal Type	Resistance Sensor
Minimum Temperature	-2.000000E+2 °C
Maximum Temperature	4.000000E+2 °C
Minimum Electrical Value	1.900000E+1 Ohm
Maximum Electrical Value	2.470000E+2 Ohm
Mapping Method	RTD
0 C resistance = 100 Ohm	1.000000E+2 Ohm
Callendar-Van Dusen coefficient A	3.908300E-3 1/C
Callendar-Van Dusen coefficient B	-5.775000E-7 1/C²
Callendar-Van Dusen coefficient C	-4.183000E-12 1/C³
Response Time	1.000000E-6 sec
Excitation Current (Nominal)	1.995262E-3 A
Excitation Current (Maximum)	1.000000E-2 A
Calibration Date	11/18/2012
Calibration Initials	NJJ
Calibration Period (Days)	365 days
Measurement location ID	0
Domain parameter of the Calibration	Electrical

Target Location

Hardware TEDS

Virtual TEDS file path

C:\Program Files\National Instruments\Shared\TEDS\nenad.ted

EEPROM

Device Family

DS_2433 0x 23

EEPROM Device Address (Only Needed for Multi-Device)

x23 x2C x69 xAB x00 x00 x00 xBC

LSB MSB

Create New TEDS

Read TEDS Data

Write TEDS Data

Fig. 8. Graphical User Interface for TEDS Editor.

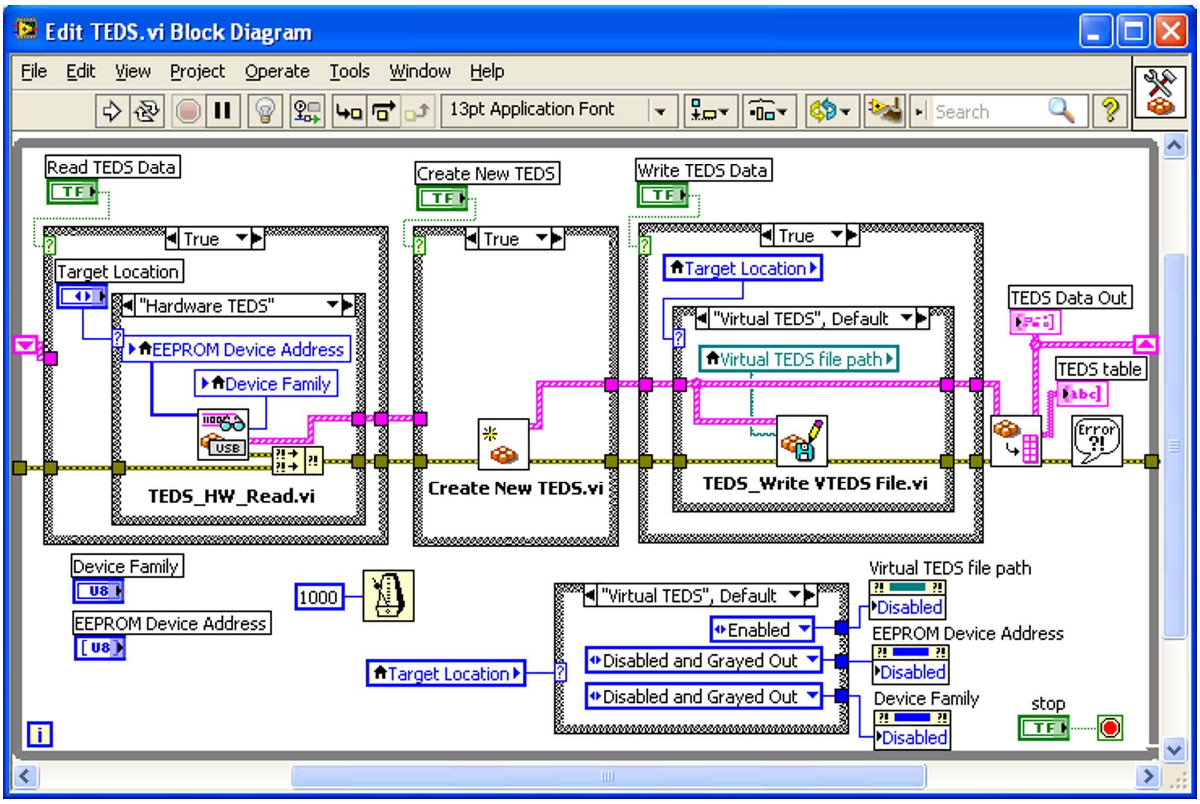


Fig. 9. Block diagram of TEDS Editor.

In the case when VTEDS is used, a sub-program reads the data from the selected TEDS file. The received data is then parsed using NI's TEDS library [24] and presented to the user in the form of a table. The first five rows of *TEDS Information* table (Fig. 8) represent basic TEDS, followed by standard template data for RTD sensor, calibration table template, and user data at the end of the table.

Create New TEDS option from TEDS editor front panel launches the second sub-VI that allows the user to create new TEDS. This sub-VI provides functions to enter new data for basic TEDS and one of eight most frequently used standard transducer TEDS templates. In addition, it allows the user to add calibration data and free form user or manufacturer data.

By selecting the option *Write TEDS Data*, the application launches the third sub-VI to access a previously discovered EEPROM device and to write TEDS data using a One-wire protocol. If VTEDS is used, TEDS data is stored in the selected local file.

5. Conclusions

This paper presents the design and implementation of a plug-and-play resistance temperature sensor. Work was focused on the process of implementation of two main components of the smart sensor: Mixed-mode transducer interface and Network Capable Application Processor. To support the realization of these key elements, appropriate hardware and software tools are developed, implemented and tested.

A mixed mode interface has been developed for connection of a platinum RTD sensor with the IEEE 1451.4 DAQ/NCAP system. It allows for transducer signals and digital TEDS data to be transferred via separate connections. To support communication over MMI, a standard 8-bit microcontroller is used. To improve measurement accuracy and to make the measurement independent of the current source and ADC offset errors, the reference resistor method has been used.

Microcontroller firmware that has been developed in C programming language, controls analog-to-digital conversion of the analog transducer signal, implements a One-wire protocol for reading and writing TEDS data, and supports USB communication functions with the PC. Micro-controller’s firmware scans and discovers connected TEDS sensors and based on the URN number of a TEDS memory device automatically adjusts the communication protocol, so the different memory devices can be used for TEDS data storage.

NCAP's application processing and network communication functions are implemented based on the concept of virtual instrumentation using a PC. On the basis of TEDS data read, the implemented PC program performs plug-and-play sensor interfacing.

To provide a simple integration of implemented smart sensors into the plug-and-play concept, a PC-based TEDS editor is developed. The TEDS editor allows the user to create new TEDS, to read or edit existing TEDS, to write TEDS data into a physical TEDS memory or to save TEDS data in the form of a virtual TEDS file.

The implemented plug-and-play sensor was tested using a series of resistors of known resistance. The test results showed that the resistance measurement error originating from the NCAP electronics does not exceed ±0.32 Ω, which means that the temperature error does not exceed ±1°C within a measuring range from -200 ^{o}C to +850 ^{o}C. It should be noted that the 10-bit resolution AD converter contributes significantly to this error, which can easily be fixed by using a higher-resolution AD converter.

The developed tools can be used to implement mixed-mode interface and Transducer TEDS, which are the essential parts of a smart analog transducer. These tools can be useful in upgrading of legacy analog sensors to smart networked sensors or for conformance test of existing sensors compliant with the IEEE 1451.4 standard.

Acknowledgements

This research was supported by the Serbian Ministry of Science and Technological Development in project TR36047 and TR32043.

References

[1] Bojko, T. (2005). Smart sensor solutions for mechanical measurements and diagnostic. *Metrology and Measurement Systems*, 12(1), 95−103.

[2] Frank, R. (2000). *Understanding Smart Sensors, 2nd ed.* Norwood: Artech House.

[3] Drndarevic, V., Bolic, M. (2008). Gamma Radiation Monitoring with Internet-Based Sensor Network. *Instrum. Sci. Technol.*, 36(2), 121−133.

[4] Wobschall, D. (2005). A Minimal Dot 4 NCAP with Compatible Sensor Bus. In *Proceedings of Sensosrs for Industry Conference*. Houston, TX, 139−142.

[5] Ramos, H. G., Postolache, O., Pereira, M., Girao, P. S. (2006). Embedding IEEE 1451.4 smart sensing node in a Wireless Air Quality Monitoring Network. In *Proceedings of 49th IEEE International Midwest Symposium on Circuits and Systems (MWSCAS)*. San Juan, PR, 177−181.

[6] Licht, T. R. (2001). The 1451.4 Proposed Standard. *IEEE Instru. Meas. Mag.*, 4(1), 12−18.

[7] IEEE The Institute of Electrical and Electronics Engineers. (2004). *1451.4 IEEE Standard for A Smart Transducer Interface for Sensors and Actuators — Mixed-Mode Communication Protocols and Transducer Electronic Data Sheet (TEDS) Formats.* New York: IEEE.

[8] Lee, K. B., Song, E. Y. (2008). Understanding IEEE 1451-Networked Smart Transducer Interface Standard. *IEEE Instru. Meas. Mag.*, 11(2), 11−17.

[9] Ulivieri, N., Distante, C., Luca, T., Rocchi, S., Siciliano, P. (2006). IEEE1451.4: A way to standardize gas sensor. *Sensor. Actuat. B-Chem.*, 114(1), 141−151.

[10] Kim, J. D., Kim, D. J., Byun, H. G., Ham, Y. K., Jung, W. S., Han, D. W., Park, J. S., Lee, H. L. (2007). The definition of basic TEDS of IEEE 1451.4 for sensors for an electronic tongue and the proposal of new template TEDS for electrochemical devices. *Talanta*, 71(4), 1642−1651.

[11] Yurish, S. Y. (2005). IEEE 1451 Standard and Frequency Output Sensors: How to obtain a Broad-Based Industry Adoption. *Sensors & Transducers Magazine*, 59(9), 412–418. *http://www.sensorsportal.com*.

[12] Liu, J., Li, Y., Zhao, H. (2010). A Temperature Measurement System Based on PT100. In *Proceedings of International Conference on Electrical and Control Engineering (ICECE 2006)*. Wuhan, China, 296–298.

[13] Svelto, C., Galzerano, G., Bava, E. (2001). Compact and accurate digital thermometer based on Anderson's loop and Pt-100 sensor. *Measurement*, 29, 287–292.

[14] DS2433 4Kb 1-Wire EEPROM. (June 2012). *http://datasheets.maximintegrated.com/en/ds/DS2433.pdf.*

[15] 1-Wire Devices. (June 2012). *http://www.maximintegrated.com/products/1-wire/*.

[16] AT90USB1286 Microcontroller. (June 2012). *http://www.atmel.com/devices/at90usb1286.aspx*.

[17] Jevtic, N., Drndarevic, V. (2012). Development of Smart Transducers Compliant with the IEEE 1451.4 Standard. In *Proceedings of 8th IEEE International Symposium on Instrumentation and Control Technology (ISICT)*. London, UK, 126−131.

[18] Evaluation of measurement data. (2008). *Guide to the expression of uncertainty in measurement*, JCGM 100, GUM 1995 with minor corrections, First edition.

[19] Dzwonkowski, A., Swedrowski, L. (2012). Uncertainty analysis of measuring system for instantaneous power research. *Metrology and Measurement Systems*, 12(3), 573−582.

[20] Atmel AVR Studio. (December 2012). *http://www.atmel.com/tools/atmelstudio.aspx*.

[21] National Instruments LabVIEW. (June 2013). *http://sine.ni.com/np/app/main/p/docid/nav-104/lang/pl/*.

[22] Agilent 34401A Data Sheet. (June 2013). *http://cp.literature.agilent.com/litweb/pdf/5968-0162EN.pdf*.

[23] Jones C. H. (December 2012). IEEE 1451.4 Smart Transducers Template Description Language. *http://standards.ieee.org*.

[24] How to Install the TEDS Library into LabVIEW. (June 2013).
http://digital.ni.com/public.nsf/allkb/DE722DD48F72175986256EFB00545EB8.