Sensor Network with Data Transfer over Power Supply Wires

Vašek V., Dolinay J. and Dostálek P.

***Abstract*—**This article describes system which allows creating sensor networks where sensors communicate with a central unit using only two wires – the power lines. The system can be used in wide range of applications in which it is needed to collect data from several sensors connected by only two wires. The wires provide power to these sensors and at the same time transfer the data.

***Keywords*—**sensor, microcontroller, power line communication, data transfer, HCS08, PWM.

I. Introduction

PROCESS measurement is one of the most important parts of any control system. It follows from the fact that control accuracy depends on how precisely the measuring chain works. These days there is number of devices available performing data acquisition tasks – standard cards for PCI or ISA bus which are suitable for standard personal computers and its industrial versions and modules for industrial automation usually equipped with RS485, CAN and other interfaces [4]. Independent category is formed by smart sensors incorporating sensor, converter to unified signal and data acquisition device in one embedded system with very compact dimensions and low power consumption. They have number of advantageous features, such as automatic diagnostic and calibration, high accuracy and immunity against electromagnetic interference due to short signal paths. Such sensors can be found in increasing number of applications, including automotive and aircraft industry, where small dimensions and low weight are crucial.

Rather often it is necessary to measure data in terrain where it is not possible to use standard computer equipped with DAQ card. In this case it is very advantageous to use laptop computer or other device with suitable communication interface equipped with portable sensor network.

The main intention for our work is to develop simple sensor network minimizing number of hardware components of the system, its configuration complexity and deploy time of the measurement system.

II. System overview

The following requirements were kept in mind when designing the system:

- Use only 2 wires which both provide the power and carry the data.
- Transfer data securely, detect and discard damaged data.
- Make the devices as cheap as possible.

The system consists of one or more sensors (transmitters) which are all connected to one two-wire line starting from a central unit (receiver). All sensors are powered from this line. Transmitting data from the sensor is done by lowering the voltage of the power line (by switching on a load in the sensor). The receiver monitors the voltage and interprets the changes as logical zeroes and ones in the transmitted digital data.

The communication is only possible in one way – from the sensor to the central unit. This could be considered a drawback, but for intended application of collecting data from e.g. temperature or humidity sensors, this one-way communication is sufficient. It has the advantage of simplicity both in software and hardware of the sensor.

We considered also master-slave model with the central unit being the master which requests data from the sensors (slaves) only when needed. This would mean better use of the bandwidth – sensors would transmit data only when requested. However, on the other hand, if we decide to require that the sensors are also able to receive data, it would require extra circuitry in each sensor which increases the price and moreover it complicates the setup of such system – address of each sensor must be defined in the central unit with the polling interval etc.

After considering all pros and cons we decided for one-way communication solution. In this solution each sensor transmits data with predefined period. This period can be fixed (e.g. 10

This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic under the Research Plan No. MSM 7088352102 and by the European Regional Development Fund under the project CEBIA-Tech No. CZ.1.05/2.1.00/03.0089).

Vladimír Vašek is with the Department of Automation and Control Engineering, Tomas Bata University in Zlin, Faculty of Applied Informatics, nám. T. G. Masaryka 5555, 76001 Zlín, Czech Republic (e-mail: vasek@fai.utb.cz).

Jan Dolinay is with the Department of Automation and Control Engineering, Tomas Bata University in Zlin, Faculty of Applied Informatics, nám. T. G. Masaryka 5555, 76001 Zlín, Czech Republic (e-mail: dolinay@fai.utb.cz).

Petr Dostálek is with the Department of Automation and Control Engineering, Tomas Bata University in Zlin, Faculty of Applied Informatics, nám. T. G. Masaryka 5555, 76001 Zlín, Czech Republic (e-mail: dostalek@fai.utb.cz).

seconds) or it can be programmed into the sensor during initial setup, together with unique serial number. In the central unit it is easy to choose the period with which the data from sensor are received and drop the extra measurements, which are not needed.

An example of the sensor network can be seen in the following picture. There are three sensor connected to a central unit, which receives and decodes the data from the sensors and passes them on to the laptop. In the laptop it is then easy to store the data, display their trends etc.

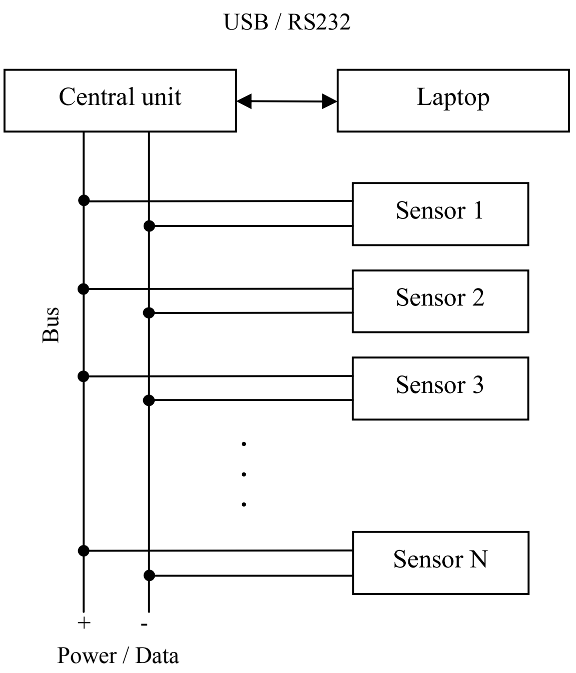

Fig.1. Sensor network block diagram.

III. Main parts of the system

The system consists of three main parts: central unit (receiver), sensor unit (transmitter) and a programming unit.

A. Central unit

This unit receives data from all the sensors and provides these data to supervisory system. In our design this is a simple MCU-based device which handles the low-level operations and presents data to a personal computer for storing, displaying etc. Basically the unit provides power source for all the sensors and it monitors the voltage of the power source to receive data from the sensors.

Of course, it would be possible to create a unit which would also present the data on display or store them to storage device. But for our purpose we prefer to keep the unit simple and let the user manipulate the data as needed in the supervisory computer. This has the advantage of bigger versatility of the unit.

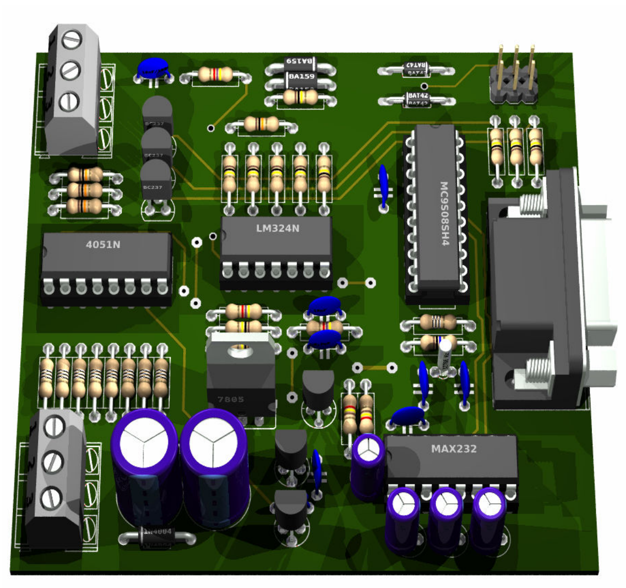

Fig. 2 Central unit hardware

The Central unit hardware can be divided into six functional blocks:

- High-pass filter – eliminates DC component (12 V supply voltage) from useful signal.
- Low-pass filter – shape high-pass filter output signal to near rectangular wave form.
- Amplifier – amplifies signal before it enters to A/D converter. Gain is selectable in 8 steps by microcontroller.
- Control logic – utilizes same microcontroller as sensor unit
- Communication interface – standard RS232 interface Power supply – provides stabilized +5 V for digital and +/-8 V for analog circuits (operational amplifier and multiplexer).

Detailed description of the unit including schematics can be found in the Hardware chapter below.

B. Sensor unit

The sensor unit is the transmitter, which transmits data to the central unit with certain period. For example, this can be temperature sensor which sends the measured temperature once every 30 seconds. In fact we can divide this unit into two parts:

- sensor itself (e.g. temperature sensor as an integrated circuit as available from semiconductor manufacturers, e.g. with PWM output)
- bus module, which handles the communication with the central unit. In our design the sensor unit is equipped with simple 8-bit microcontroller which handles both the communication on the bus and with the sensor itself.

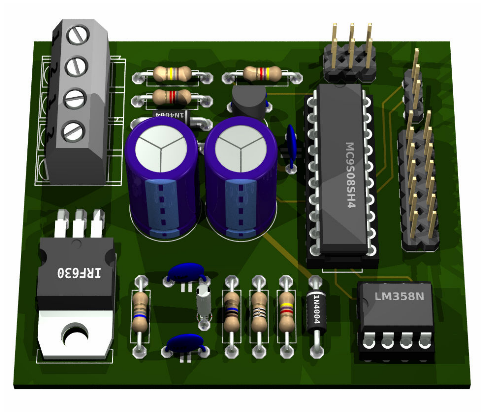

Fig. 3 Sensor unit hardware

Three sensor types are supported:

- sensors with analog output,
- sensors with PWM (or other digital) output,
- sensors with serial interface (SPI, IIC).

This allows connecting wide range of available sensors to the network. For example, the analog interface can be used to connect sensors with voltage (e.g. 0-10V) or current output (e.g. 4-20 mA). The interface for connecting digital (PWM or other) signal allows connecting sensors smart sensors, such as pre-calibrated temperature or humidity sensors. The serial interface then allows connecting devices with IIC and SPI interface, which includes even accelerometers, gyroscopes and so on.

C. Programmer unit

The programmer unit is device which is used for initialization of the sensor units. It is used only when the sensor network is set up or re-configured. It is not needed for normal operation of the system.

The sensor units are assumed to have their microcontrollers programmed before assembly (soldering them to the PCB) and given the overall price of the unit it is not necessary to allow for firmware change in the microcontroller. However, it is possible to store some configuration data into the MCU, such us unique serial number of the sensor, period of transmission and possibly also other parameters. The programmer unit makes it possible to write these data into the sensor unit.

The programmer unit is a simple MCU-based device, which is connected to the sensor unit to be programmed. It is also connected to a PC, which provides user interface for setting up the parameters (there is no interface on the programmer unit itself). The PC can provide additional services, for example, store the database of sensor IDs.

IV. COMMUNICATION PROTOCOL

As mentioned above the sensor units transmit data to the central unit by the means of lowering the voltage of the power source, which is monitored in the central unit and decoded as logical ones and zeroes. It uses PWM encoding with period of 2 ms; the pulse is the voltage drop. Whether this drop is interpreted as one or zero depends on its length. Nominal length of 0.8 ms represents 1, length 0.35 ms represents 0, see figure below. The software allows for inaccuracy in the length up to certain limits – these limits were experimentally set to +/- 0.1 ms. Voltage drops of a length which does not fall into the tolerance are considered a noise and ignored.

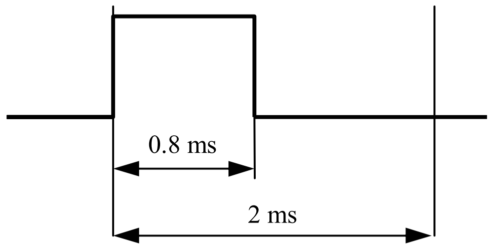

Fig. 4 – The pulse for log. 1

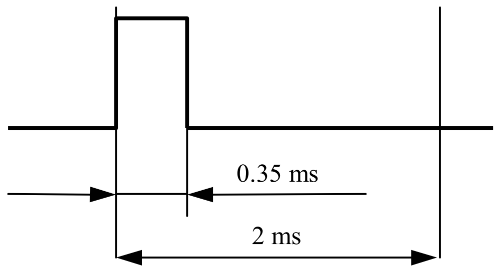

Fig. 5 The pulse for log. 0

Data are sent in packets. Each packet begins with 3 ms start pulse, delay of 2 ms, then transmission of the data bits with 2 ms period. The structure of the data can be seen in Fig. 6.

1 Byte	*1 Byte*	*N Bytes (max. 255)*	*1 Byte*
Sensor ID	Length	Data 1, Data 2, …, Data N	CRC 8

Fig. 6 data packet structure

There is 1 byte for the ID of the sensor, one byte with length of the data, the data itself and then 1 byte CRC. The CRC is used for detection of errors including collision (if more than one transmitter transmits at the same time).

V. HARDWARE OF THE SYSTEM

Hardware of the system can be divided into the three types of units: sensors transmitting data to the bus, central unit receiving data from the bus and finally programmer unit which is used to program configuration data to the sensor using simple TTL UART interface.

Sensor and central unit hardware design is based on 8-bit microcontroller Freescale MC9S08SH8. It is a member of low-cost, general purpose, high-performance 8-bit flash-based microcontrollers with Von-Neumann architecture. Central processor unit with enhanced HCS08 core is fully upward compatible with Freescale HC05 family. CPU architecture is fully optimized for C language compilers. On the chip are integrated many of modules, for example [1]:

- On chip 8 KiB FLASH memory with in-circuit programming capability
- 512 B on-chip RAM
- Up-to 40 MHz CPU clock speed
- Internal clock generator
- Two 2-channel 16-bit timer/pulse width modulator modules (TPM)
- Serial communication interface (SCI)
- Serial peripheral interface (SPI)
- 10-bit Analog-to-digital converter with 12-channel analog multiplexer
- 17 general-purpose I/O pins, 1 output only
- Watchdog system
- and other modules, see [1].

A. Central unit

Central unit utilizes same microcontroller type as sensor unit. Its hardware can be divided to six functional blocks:

- High-pass filter and over-voltage protection
- Low-pass filter
- Amplifier
- Control logic
- Communication interface
- Power supply

Bus voltage first enters high-pass filter with over-voltage protecting circuit eliminating DC component from useful signal. Then signal shape is corrected in the second order Butterworth type active low pass low-pass filter with Sallen–Key topology implemented by operational amplifiers (Fig. 9). Filter parts was designed for cutoff frequency of 5000 Hz and gain of 0 dB in the passband.

Parts values were designed using procedure published in [7]. Computation is based on transfer function of the Sallen-Key 2nd order low-pass filter (1) where coefficients a_1 and b_1 are equal to (2) and (3).

$$A(s) = \frac{A_0}{1 + a_1 s + b_1 s^2} \qquad (1)$$

$$a_1 = \omega_c \left[C_1 (R_1 + R_2) + (1 - A_0) R_1 C_2 \right] \qquad (2)$$

$$b_1 = \omega_c^{\,2} R_1 R_2 C_1 C_2 \qquad (3)$$

Final transfer function of the 2nd order low-pass Sallen-Key filter is:

$$A(s) = \frac{A_0}{1 + \omega_c \left[C_1 (R_1 + R_2) + (1 - A_0) R_1 C_2 \right] s + \omega_c^{\,2} R_1 R_2 C_1 C_2 s^2} \qquad (4)$$

where ω_c is a cutoff angular frequency, A_0 is gain of the filter in the passband and a_1 and b_1 are filter coefficients determining its properties. After the formulation of R_1 from equation (3) and constituting to (2) we obtain quadratic equation:

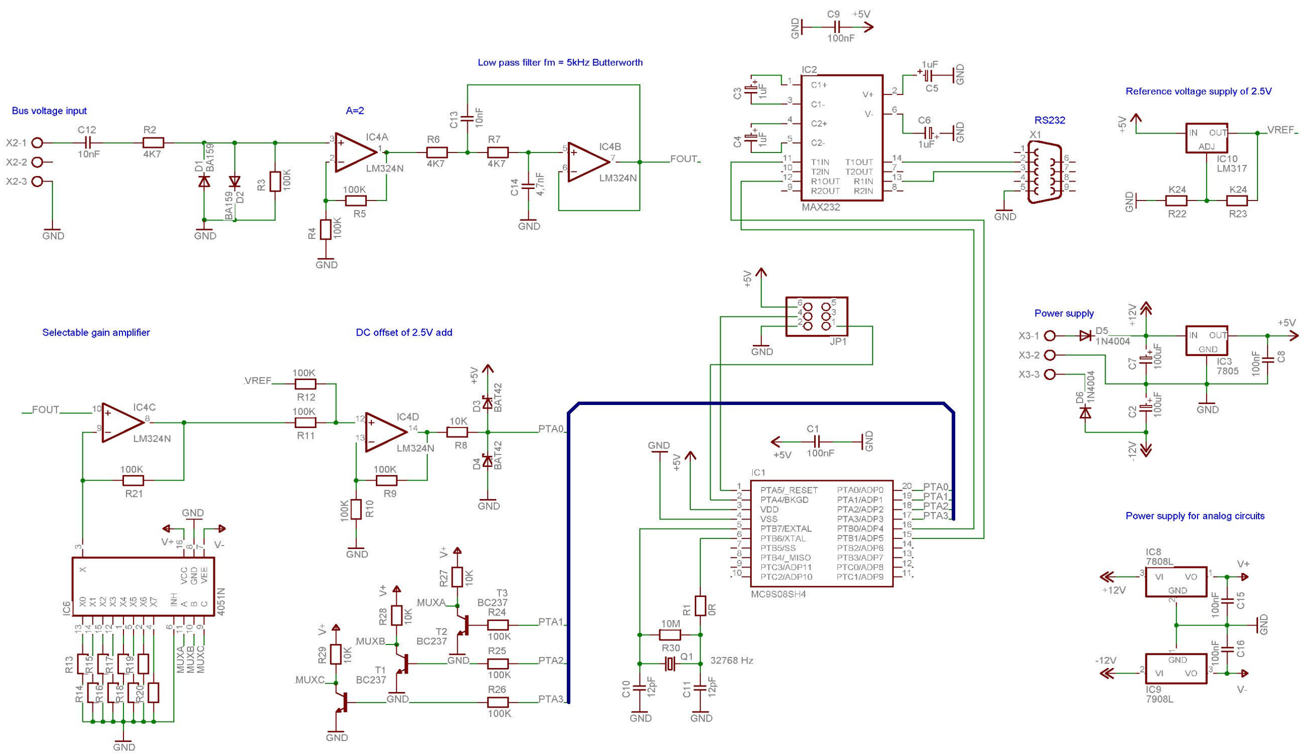

Fig. 7 Complete schematics of the central unit

$$R_2^{\,2}C_1^{\,2}C_2\omega_c^{\,2} - a_1R_2C_1C_2\omega_c + b_1\left(C_1 + C_2 - A_0C_2\right) = 0\,. \quad (5)$$

Its solution is equation for computation of R_1 part value (6), R_2 part value can be computed by (7).

$$R_1 = \frac{b_1}{R_2C_1C_2\omega_c^{\,2}} \quad (6)$$

$$R_2 = \frac{a_1C_1C_2\omega_c + \sqrt{\left(-a_1C_1C_2\omega_c\right)^2 - 4C_1^{\,2}C_2\omega_c^{\,2}b_1\left(C_1 + C_2 - A_0C_2\right)}}{2C_1^{\,2}C_2\omega_c^{\,2}} \quad (7)$$

In order to obtain non-negative value under square root in (7), capacitor values must fulfill (8).

$$C_2 \geq C_1\frac{4b_1A_0 + a_1^{\,2}A_0 - a_1^{\,2}}{a_1^{\,2}A_0} \quad (8)$$

Practically the easiest way is to choose first capacitors C_1 and C_2 manufactured usually in E6 series and then compute resistor values. Exact resistor value can be reached by connecting more resistors in parallel or in series.

After amplification to the suitable level in the selectable gain amplifier (Fig. 8) consisting of parts IC4C and IC6 is signal directly brought to the analog input of the microcontroller. Communication interface is in basic version standard RS232 realized by TTL to RS232 level converter MAX232 (Fig. 10) connected to MCU UART interface. There is an option to use USB interface utilizing USB UART interface FT232BM integrated circuit [3]. In this case computer detects central unit as standard USB serial port device.

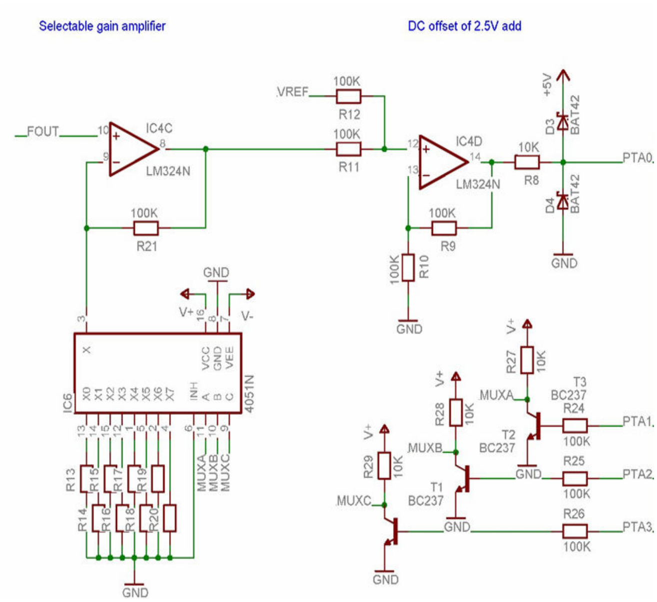

Fig. 8 Selectable gain amplifier

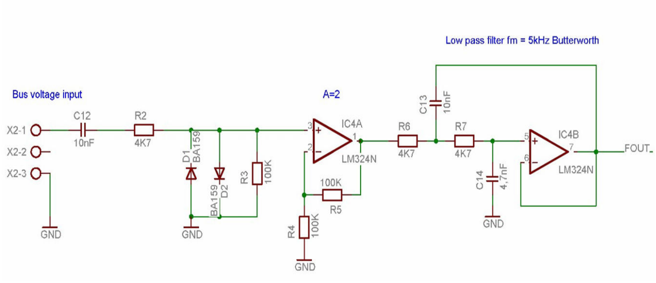

Fig. 9 high-pass and low-pass filters

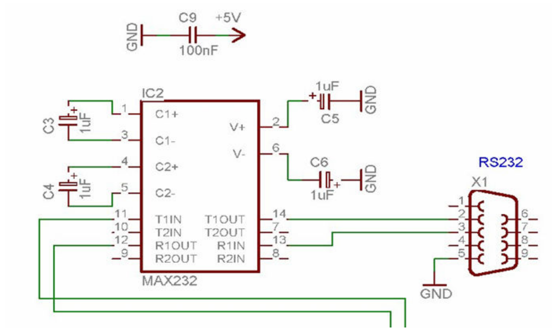

Fig. 10 Communication interface

B. Sensor unit

Sensor unit hardware can be divided to the three functional blocks:

- Power supply
- Modulator
- Control logic

The complete schematics of the sensor unit can be seen in the Fig.12.

Input voltage from the bus enters power supply circuit which is based on three terminal positive voltage regulator 78L05 with output voltage of 5 V and 100 mA output current capability. It is used for supply digital parts such as microcontroller and sensor connected to sensor unit I/O interface. Diode D1 connected before regulator protects sensor units from polarity reversal and separates filtered supply voltage from the bus too. Without this diode it would be impossible to effectively modulate bus voltage when data transfer takes place due to high-capacity filter capacitors in each sensor module. From the non-stabilized filtered bus voltage is supplied operational amplifier LM358 which is used in modulator and bus voltage sensing feedback circuit. All sensor unit functions are controlled by microcontroller Freescale MC9S08SH8 (IC1). System clock is generated by Pierce crystal oscillator with output frequency of 32.768 kHz (Q1) connected to EXTAL and XTAL pins of the MCU. External reference clock is by internal MCU's PLL circuit increased to 40 MHz resulting in internal bus clock frequency of 20 MHz. Modulator circuit consists of operational amplifier IC3A and MOSFET IRF630 (T1) acting as constant current load switched with PTC3 pin of MCU to on or off state leading to the bus voltage variations when data are transmitted. Before and during data transmit MCU is sensing bus voltage which is to appropriate level for A/D converter adapted using voltage divider R5, R6, R7 and operational amplifier IC3B. Bus voltage sensing is used for collision detection and monitoring of modulator function. Sensor is connected to the 14 pin sensor I/O interface SV2 providing two analog inputs complete SPI interface (signals MISO, MOSI, SPSCK), two general purpose digital inputs and outputs and regulated 5 V supply voltage. Sensor unit configuration can be updated by TTL UART interface connected to 3 pin pinheader SV3. BDM interface connector SV1 is used for device firmware updating respectively for debugging in SW development stage.

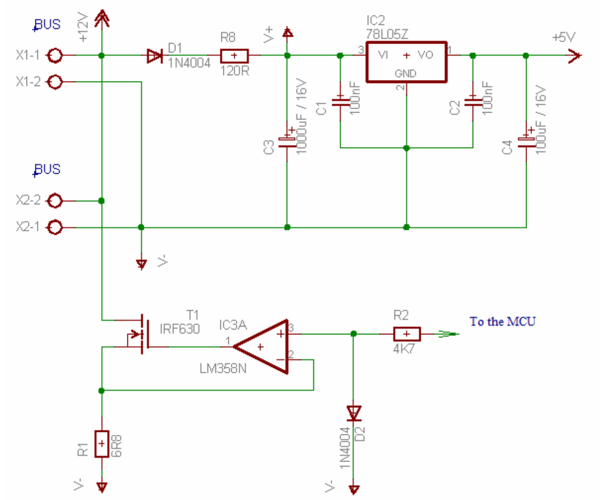

Fig. 11 Power supply and modulator schematics

VI. Software of the System

The software consists of programs for the MCUs in the central unit, sensor unit and programmer (embedded software) and the programs for PC which allows communication with the central unit and the programmer unit (PC software).

A. Embedded Software

The embedded software is written in C language. It is developed in Freescale Code Warrior IDE, which is the development tool provided by Freescale for their microcontrollers.

First version of the program is implemented in plain C language but for future development we intend to use real-time operating system RTMON [6] to make the development easier even with more features implemented in the program.

The program for the transmitter performs the following tasks (see the figure below):

- Check if there is a request from the programmer unit for setting up the parameters of the sensor unit. If there is such a request the unit enters special programming mode in which it receives configuration parameters from the programmer unit and saves them into internal flash memory. This communication is carried out via serial interface (UART).
- During normal operation the transmitter unit reads data from the sensor. The exact way of doing this depends on the type of the sensor which is connected. This type of sensor is defined by parameter in the device memory. For example, if the connected sensor has analog output, reading the data involves performing analog-to-digital conversion on the built-

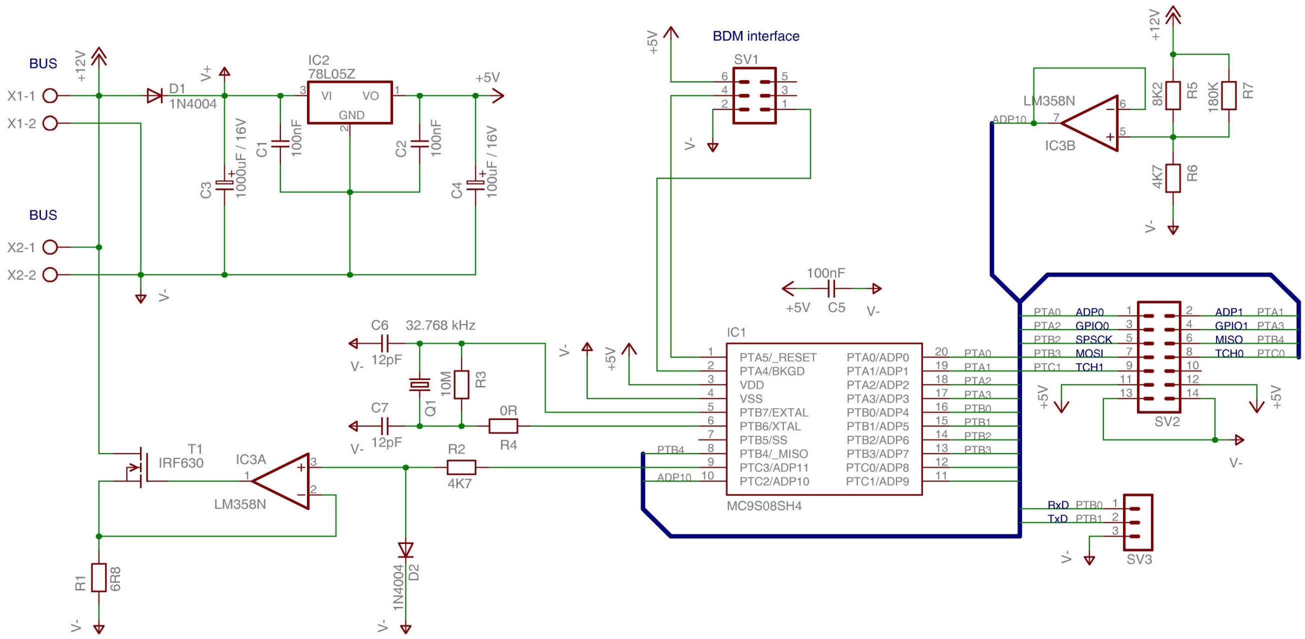

Fig. 12 Complete sensor unit schematics

in AD converter of the MCU. If the sensor type is PWM or other digitally coded type, reading data involves processing the digital input on the MCU input pin.

- The data from the sensor are then filtered and transformed to required range (as defined by the configuration parameters).
- Last part of the process is transmitting the data to the central unit. This is done using PWM modulation as described in the chapter about communication protocol. In principle the MCU uses its output pin (binary output) to open MOSFET transistors which applies load to the power supply and thus creates a small voltage drop on the line.

The program for the central unit (receiver) continuously monitors the voltage on the power line. The hardware of the receiver amplifies the voltage differences and provides this amplified voltage to the AD converter in the microcontroller. The MCU samples the analogue input and detects the pulses which encode the data.

The flowchart of evaluation of the pulses can be seen in figure below. It depicts the part of the program which validates the received pulse (this is done in interrupt handler). If the pulse is valid, i.e. if its length falls into the limits set for logical 0 or logical 1; it means new bit has just been received. This bit is stored into a variable at the appropriate position, which depends on the received bits count. Once whole byte is received, it is passed to the main loop of the program for further processing. If at any time invalid bit is received (i.e. there is a pulse detected with length which does not fall into limits for neither logical 0, nor logical 1), the received bit count is reset and also the program resets it internal state and waits for new start pulse. This means the whole packed is discarded.

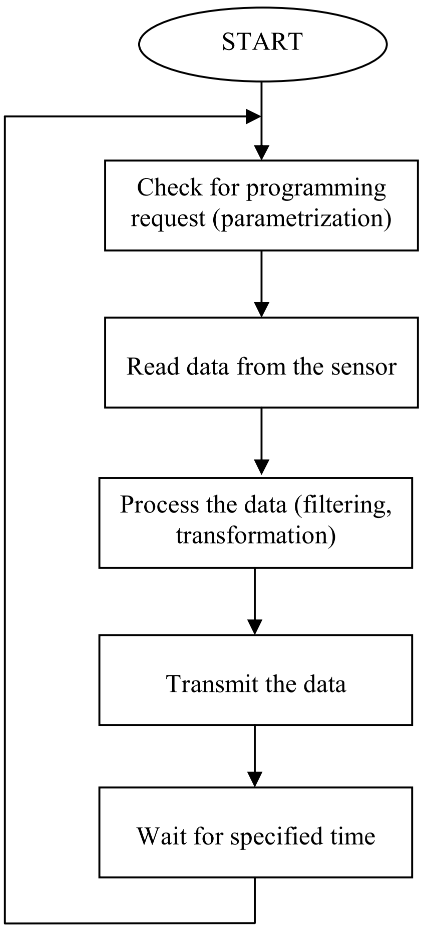

Fig. 13 Flowchart of the transmitter program

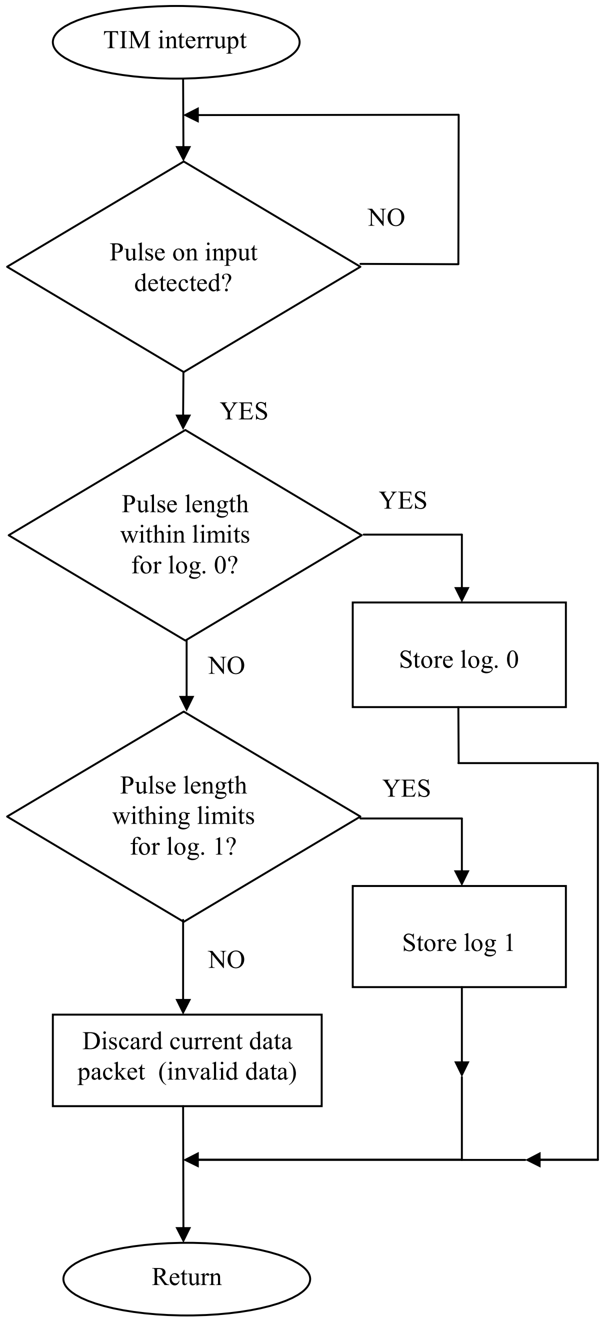

Fig. 14 Flowchart of the receiving interrupt handler

B. PC Software

The PC software is still under development. It is planned that there will be graphical user interface for setting up the network and for setting the parameters in the sensor units. This program will allow for preparing the network and for basic visualization and storage of the data obtained from the sensor. For processing the data by 3rd-party programs there should be open interface (e.g. a DLL library or some communication protocol) which will allow users to use the data in their own programs. We also plan to create drivers for commonly used tools such as Matlab or Control Web, which will allow easy integration of this sensor network with these programs.

Currently a terminal program is used on PC for communicating with the central unit and programmer. The standalone GUI applications which will allow more comfort in controlling the units should be implemented in future.

VII. Conclusion

This paper described system for collecting data from sensors using just two wires. The same wires are used both for powering the sensors and for communication between the sensors and the central unit which collects the data.

The system can be used in applications which require monitoring several quantities in a number of places, such as, for example, monitoring the temperature and humidity in a building or monitoring various quantities in a technological process. The components of the system are based on 8-bit microcontrollers, which allow them to be low-cost and simple to implement. Currently, the system development is still in progress, more features will be added in future.

References

[1] Freescale Semiconductor. MC9S08SH8 Data Sheet rev.3, 2008. Available from WWW: <www.freescale.com>, Accessed: 2011-07-12.

[2] Freescale Semiconductor. CPU08 Central Processor Unit, 2001. Available from WWW: <www.freescale.com>, Accessed: 2011-07-12.

[3] Future Technology Devices International. FT232BM USB UART IC Data Sheet, 2010. Available from WWW: <www.ftdi.com>, Accessed: 2011-07-12.

[4] Dolinay, J., Dostálek, P., Vašek, V., Kolomazník, K. & Janáčová, D. Modernization of Control System for Enzymatic Hydrolysis, WSEAS Transactions on Systems and Control, 2009, pp. 799-808.

[5] Morton, T. D., Embedded Microcontrollers, Prentice Hall, 2001.

[6] Dolinay, J., Dostalek, P., Vasek, V., Implementation and Application of a Simple Real-time OS for 8-bit Microcontrollers, In procceedings of the 10th WSEAS International Conference on Electronics, Hardware, Wireless and Optical Communications (EHAC'11), Cambridge 2011, pp.023-026, ISSN1792-8133.

[7] Texas Instruments. Op Amps for Every One - Design Reference [online]. 2002. [cit. 2011-07-24]. Available on WWW: <http:// www.ti.com>.