

## Energy Sub-Metering Data Acquisition System

**V. Bulbenkiene, A. Pecko**

*Department of Informatics Engineering, Klaipėda University,  
Bijūnų str. 17, LT - 91225 Klaipėda, Lithuania, phone: +370 46 398986; e-mails: bulbenkiene@gmail.com,  
aleksej.pecko@gmail.com*

**E. Zulkas, A. Kuprinavicius, A. Sokolov**

*Department of Informatics, Klaipėda University,  
H. Manto str. 84, LT-92294 Klaipėda, Lithuania, phone: +370 46 398820; e-mails: zulkas.e@gmail.com,  
kuprinavicius.a@gmail.com; sokolov.lex@gmail.com*

**G. Mumgaudis**

*JSC Koris,  
Danės str. 6, LT-92109 Klaipėda, Lithuania, phone: +370 698 76522, e-mail: gm@koris.lt*

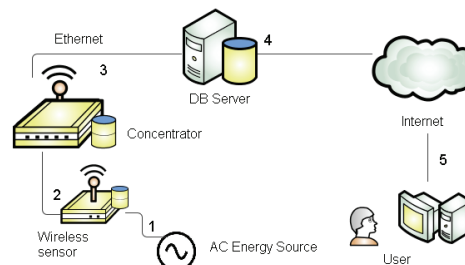
### Introduction

Increasing energy consumption and rising energy prices causes effective use of energy to become more and more important. To fulfill this objective, energy monitoring systems are being created which allows data collection and display the consumption of resources [1–3]. Energy sub-metering system usually consists of three main parts: the energy metering unit, data transmission system to transmit data to various data repositories and a system that graphically displays the data received or generates data reports. Data collection begins with the physical energy data measurements. For example, gas and water consumption data can be read optically. Electricity consumption can be measured using two methods: by scanning voltage and current instantaneous values with sufficient speed [4], or without the intervention to the electrical circuit using electrical sensors to read only the instantaneous current values [5–6]. To develop a prototype sub-metering system the second energy accounting method that allows the system to automatically monitor and collect data on energy consumption was chosen. Monitoring results must be easily accessible to the user not only on the Web, but also by the alternative users chosen means of communication (by fax, email, SMS). This paper analyzes the non-standard error processing solutions and their application to create a prototype, also issue-specific solutions related to energy transmission and processing using standard wired and wireless transmission protocols.

### Main data collector

After reading the data, it is transferred to primary storage (Fig. 1, stage 1). Filtered data is sent to the Meter-

Bus module which broadcasts the data to the concentrator (Fig. 1, stage 2). Energy consumption data is transferred together with service information which allows addressing and managing the terminal nodes. Data is provided to the concentrator's CPU which stores data in external memory. Depending on needs the system can use an external EEPROM or flash memory. In this work the choice is the EEPROM memory. When certain amount of data is collected it is sent from the concentrator to a remote SQL database via the Internet (Fig. 1, stage 3). The user can view a limited history of consumption data at the concentrators LCD display. User also has the opportunity to graphically review all past data according to the chosen time interval or generate energy consumption reports. These services are provided to the user connected to HTTP server on the Internet (Fig. 1, stage 5).



**Fig. 1. Energy sub-metering system**

### Software code implementation in energy monitoring system

One of the most complex tasks in communication protocols are non-standard situations, exceptions and errors caused by interference between the individual system components (such as SQL Server) uncertain availability

and the processing. For example, at any time access to the server might be interrupted and data validation reply will not be received, although data was transmitted. This situation must be dealt with: after a certain period of time an attempt should be made to connect again and re-upload the data. If the server receives two identical data packets, one of them must be rejected. Such non-standard situations might be very different and without appropriate treatment system might not operate smoothly in real conditions of system use. Because of that in system design special attention was devoted to this problem. The concentrator must simultaneously handle a lot of tasks: obtain energy consumption data from Meter-Bus interface, write data to the EEPROM which takes a lot of time to complete, output information to LCD display, access the database, do some real-time computing, reading physical buttons press states. To ensure smooth performance of all tasks it is necessary to use one of the tasks management techniques [7]: cooperative task management, foreclosing task management, cooperative management with the ability to pause the task. It is also possible to use existing RTOS (real-time operating systems). For the system creation, cooperative task management is chosen. Based on this method a subsystem structure was developed and this structure is illustrated in Fig. 2. Its core - *task manager engine()* function which is invoked while processor is idle. Tasks are planned using *schedule\_task()* function. Function *shedule\_task()* allows to set task parameters and set the time delay before the task start. After the specified time interval, the task is placed in the queue and at the first possible processor idle state, task is executed.

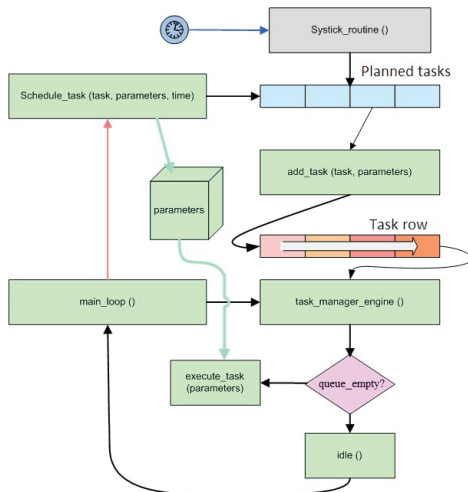


Fig. 2. Cooperative task management subsystem

While executing large amount of tasks, functions should not be allowed to operate for a long time. It is hard to write a function so that it would require little CPU time because data transmission and writing takes a long time. For example sending of one byte using the UART transmission protocol with the Meter-Bus module lasts for 0.5 ms at 19200 bits per second, and one CPU instruction is performed in 0.05  $\mu$ s. This means that in the transfer time of one byte, 10000 instructions can be performed. So it is necessary to use the interrupt mechanism. If the program runs asynchronously, there is a consistent

program execution and the outer queries processing problems. While developing a prototype sub-metering system special attention was given to the implementation of finite-state machines.

By using the microcontrollers interrupt mechanism a problem of data transfer to an external function arises [7]. Sometimes it is possible to process the data received within the interrupt, but when the operation takes a relatively long period of time there is an additional problem: during the operation of an interrupt handler another interrupt cannot be started and there is a need to wait for the CPU to return from the running interrupt. When the maximum possible interrupt delay time is less than the maximum length of any interrupt handler execution time it is impossible to guarantee adequate response time and the program may not function properly resulting in data loss. In this case it is necessary to take a part of the algorithm from the interrupt handler to the main program. Ideal would be to leave only the necessary code in the interrupt handler for example reading data from the registry or storing the byte in the output register or perform other tasks.

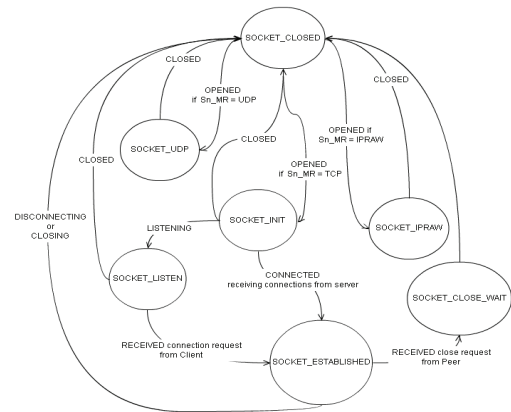


Fig. 3. TCP/IP sockets finite-state machine diagram

Algorithms associated with the further processing of the data must be written outside of interrupt handler. The fact that an interrupt was performed can be transmitted through the global flag variables. The operation of the program depends not only on the current event but also on an event from the previous state, that's why there is a need to use finite-state machines. A finite-state machine is designed to enable monitoring and controlling system states change depending on previous states and input signals (Fig. 3). This allows the states to be changed in interrupts and the main finite-state machine processing program. Actions that are associated with states can be carried out in a more preferred location of a program. There can be a lot of related and unrelated finite-state machines in the program. Using finite-state machines can greatly simplify the writing of the code and create a clearly visible structure. It is possible to realize a sufficiently complex application functionality (such as HTTP query processing, radio telegram transceiving, etc.) using only one call to the finite-state machines procedures in the main program loop for each function.

While programming microcontrollers, time intervals can be calculated using built in timers. Timers can be set to frequency divider; some timers can be set to a maximum comparable value. When the maximum comparable value

cannot precisely fit to task, timer can be set to initial value on overflow interrupt or at time interval calculation beginning. ATmega128 first and third timers can be set to 3 different comparison interrupts. They can run intervals up to  $2^{16}$  TCLK clocks. TCLK can be obtained from the core processor frequency determined with a flexible divider. When calculating the maximum timer interval is not enough to calculate specific amount of time, it is possible to use the programmable Timer Interrupt counter. LCD display is used to display the data and change concentrator settings. According to the relationship between price and functionality our choice is 128x64 pixels monochrome graphic display. LCD display was chosen with ks0108 controller. It has a non-standard management interface, but it is well documented, making it easy to use. For this display an easy to use library was developed. There was also a menu designed for changing the display settings, concentrator settings and data selection. A 128x64 pixel display can easily represent up to 100 data points graph. After representing one-minute of data as one point, 100 data points corresponds to 1 hour and 40 minutes. Buttons can change data display scale, which extends monitoring capabilities.

### Wireless data transfer protocol

In order for an energy monitoring system to be quickly installed and prepared for operation a wireless data communication should be created between sensors and the concentrator. There are standard wireless data communication protocols that are designed for power monitoring [8]. It was decided to choose from two wireless protocols that are most widely used and adapted to transmit energy data.

ZigBee and Wireless Meter-Bus protocols have their advantages. Both support multiple clients at the same time and have a great operating distance (up to 1500 m). Both protocols have energy-efficient hardware implementations. Meter-Bus was developed for energy consumption data transfer while ZigBee is a more versatile and complex protocol and therefore requires more hardware resources for realization and management. Meter-Bus is more prevalent in Europe and is applied in Lithuania in accordance with the applicable standards. ZigBee is widely spread in the Americas market. Given the last two factors Wireless Meter-Bus protocol was chosen, but a possibility of adapting to the ZigBee is also considered.

Wireless Meter-Bus protocol has a flexible structure allowing the transfer of variable length and different meaning data. The packet length is sent together with the data. This simplifies the processing of the data. The service block includes two unique manufacturer ID's to avoid compatibility with other manufacturers equipment problems. Four unique device ID's are also sent and this theoretically allows one manufacturer to address 4 billion devices. For the implementation of Wireless Meter-Bus, AMB8425-M modules were chosen.

Wireless modules have the ability to automatically add packet header that is stored in memory which is not reliable on memory dedicated for energy consumption data. The control module calculates the CRC for each data block ensuring the reliability of data transmission.

### Connection with database

The link between the concentrator and the database server (Fig. 1. stage 3) is realized using an Ethernet TCP/IP protocol and a Wiznet 5100 network controller. While using an Ethernet connection protocol it is necessary to programmatically implement the finite state machine. All possible states and possible shifts from one state to another are defined.

For example the login request that was sent to the server should wait for *sock\_established* status which means the approval of login, and then the data can be sent. While response is pending other tasks should be executed. This is realized in cooperative task management method. Wiznet driver combines a low-level and high level network microcontroller management. Most attention is focused on the socket API application interface.

Network controller with reduced TCP/IP protocol stack kernel is sufficient to create an HTTP server. General purpose concentrator must operate in a server mode and make a socket slot for a client to connect. When the client sends an HTTP request, the server must send the site's content. Such a mechanism is possible through the TCP sockets with HTTP headers processing. This HTTP server implementation is adapted to use for setting concentrator parameters.

### Data storage buffers

Energy consumption data that is sent from the terminal nodes via Wireless Meter-Bus must be stored in the concentrator. Stored data must be grouped: a lot of bytes to store information cannot be allocated because of limited flash and operating memory resources, although to store more data is an advantage for such systems. Each energy resource measurement does not include date of the measurement. Data is grouped into blocks of 15 measurements together with date value of first taken measurement.

Data exchange without any intermediate buffering system has disadvantages. Microcontroller which is receiving data has one byte buffer, and if data inside the buffer is not yet processed because processing is not fast enough it will be overwritten by a new value. Data exchange systems which have such problems can be enhanced programmatically by implementing block buffer data structure.

Block data structure consists of two internal substructures:

- To navigate between blocks;
- To store the data blocks.

The substructure for navigating between blocks is made up of a maximum number of blocks, links to the first and last block and a link to a block that sends data.

In order to describe the navigation structure between blocks in C language there is a need to use the references to real-time blocks and unsigned char type block counting meter.

The data block structure is a doubly-linked list which consists of a time variable, data array and amount of data in a block counter and a reference to previous and next blocks.

## Future perspectives

The connection between the concentrator and the database is implemented by using the TCP/IP protocol. Currently a database server computer is running a background program that transforms the TCP/IP packets to information giving data.

In future development of the system it is planned to send data directly from the concentrator to the database without the use of a background program.

After the modification of the background program, it will be possible to transmit data to not only by using the TCP/IP protocol but also by using other standard data transmission protocols.

This system is intended to be applied in the future not only for electricity consumption monitoring but also for other energy resources such as gas and water.

## Conclusions

A monitoring system prototype that is capable of working automatically and does not require external human action was developed. The system allows to store and display data about energy resources consumption.

A cooperative task management subsystem for handling non-standard situations when components of the systems are not able to interact and for handling errors that are caused by disturbances was implemented.

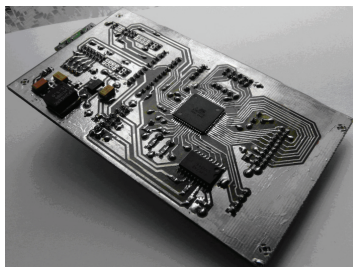


Fig. 4. Prototype of the concentrator

A concentrator (Fig. 4) with a pipeline principle of task management has been developed which allows the system to perform multiple tasks at a time.

Wireless data transmission was implemented by using AMB8425-M modules, wireless protocols and finite-state machines in order to ensure convenient installation and usage of the system.

## References

1. **Pamakštis V., Ragauskas A., Kaluževičius J., Petkus V., Bagdonas A., Borodičas P.** Advanced Energy Metering and Control System // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2008. – No.7. – P. 93–98.
2. **Yu Zhang, Musheng Yang, Zhimin Jia, Xiaoqun Cao.** Study on real-time monitoring system based on configuration software for production of infusion solutions. // *IEEE International Conference Control and Automation (ICCA'2009)*, 2009. – P. 1479–1480.
3. **Lu Ren-gui, Pei Lei, Ma Rui, Wei Jun-lei, Zhu Chun-bo.** EV energy storage monitoring system based on distributed data acquisition. // *Vehicle Power and Propulsion Conference (VPPC '09)*. – 2009. – P. 1386–1389.
4. **Bielskis A. A., Andziulis A., Ramašauskas O., Guseinovicė E., Dzemydienė D., Gricius G.** Multi-Agent Based E-Social Care Support System for Inhabitancies of a Smart Eco-Social Apartment // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2010. – No. 8(104). – P. 11–14.
5. **Gokmen G., Tuncalp K.** The Design of a Hall Effect Current Transformer and Examination of The Linearity with Real Time Parameter Estimation // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2010. – No. 5(101). – P. 3–8.
6. **Rafajdus P., Bracinić P., Hrabovcová V.** The Current Transformer Parameters Investigation and Simulations // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2010. – No. 4(100). – P. 29–32.
7. **Dodiu E., Graur A., Gaitan V. G.** Hard-Soft Real-Time Performance Evaluation of Linux RTAI Based Embedded Systems // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2010. – No. 4(100). – P. 51–56.
8. **Ahmad, R. B. Mamat, W. Juhari, M. Daud, S. Arshad, N.W.** Web-based wireless data acquisition system using 32bit single board computer. // *Computer and Communication Engineering (ICCCE'2008)*, 2008. – P. 777–782.

Received 2011 02 14

**V. Bulbenkiene, A. Pecko, E. Zulkas, A. Kuprinavicius, A. Sokolov, G. Mumgaudis.** Energy Sub-Metering Data Acquisition System // *Electronics and Electrical Engineering*. – Kaunas: Technologija, 2011. – No. 5(111). – P. 99–102.

This paper describes the creation of a system that acquires building energy sub-metering data. The system is designed in a way that allows it to use different sensors for different energy resources sub-metering. A wireless communication standard that allows to achieve the best combination of communication range, low power sub-metering and to give the system more flexibility is chosen. This paper shows how electrical energy sub-metering data is extracted by the sensor node from the current wire without any physical invasion and wirelessly passed to the concentrator. Non-standard error processing solutions and their application to the prototype, and issue-specific solutions related to energy transmission and processing using standard wired and wireless transmission protocols are analyzed. Il. 4, bibl. 8 (in English; abstracts in English and Lithuanian).

**V. Bulbenkienė, A. Pečko, E. Žulkas, A. Kuprinavičius, A. Sokolov, G. Mumgaudis.** Energijos monitoringo duomenų surinkimo sistema // *Elektronika ir elektrotechnika*. – Kaunas: Technologija, 2011. – Nr. 5(111). – P. 99–102.

Darbe aprašoma, kaip kuriama elektros energijos išteklių monitoringo sistema. Sistema sukurta taip, kad atiteityje skirtingų energijos išteklių monitoringui galėtų naudoti įvairius sensorius. Buvo pasirinktas bevielio ryšio standartas, tinkamiausias užduočiai įvykdyti pagal ryšio atstumą, energijos suvartojimą ir naudojimo lankstumą. Darbe pateikiamas elektros energijos monitoringo duomenų surinkimas srovės jutikliu, be fizinio sąlyčio su matuojamos elektros grandine. Nagrinėjamas bevielis energijos išteklių duomenų persiuntimas į koncentratorių. Realizuotos nestandartinių situacijų ir klaidų, atsirandančių dėl trikdžių ir galimo atskirų sistemos prototipo dalių neužtikrinto pasiekiamumo, apdorojimas, sudarius kooperacinio užduočių valdymo posistemę. Il. 4, bibl. 8 (anglų kalba; santraukos anglų ir lietuvių k.).