SIB: SENSOR INSTRUMENTATION BUS FOR POWER AND ENERGY CONTROL UNITS

Uroš Platiše, Mihael Mohorčič

Institut "Jožef Stefan", Ljubljana, Slovenija

Key words: SIB, MAC, MSG, Protocol, Layer, Sensor, Power, Energy, Management, Monitoring, Control

Abstract: Power and energy management in buildings, be in households, offices or in industry, is becoming increasingly important for cost as well as for carbon footprint reduction. However, in addition to power and energy management algorithms, high availability and redundancy remain key system requirements. In this paper we present a novel low cost instrumentation bus for modular system architectures, considering modular units as distributed sensors and actuators. In particular, as a show case we apply the proposed instrumentation bus for Power and Energy Control (PEC) units, providing modularity, general connectivity, high availability, scalability and future system upgradeability. Drawing analogy with sensor and actuator networks the proposed instrumentation bus also guarantees very low protocol overhead and thus maximizes the throughput of the bus.

SIB: Senzorsko instrumentacijsko vodilo za modularno krmiljenje električne moči in energije

Kjučne besede: SIB, MAC, MSG, protokol, plast, senzor, moč, energija, upravljanje, nadzor, krmiljenje

Izvleček: Upravljanje in nadzor energije v gospodinjstvih, poslovnih objektih in industriji postaja vse pomembnejše, tako z namenom zmanjševanja stroškov kot tudi emisij CO_2. Poleg algoritmov upravljanja z energijo ostajata visoka dostopnost sistema in redundančnost zelo pomembni lastnosti sistema. V tem članku predstavljamo nov koncept instrumentacijskega vodila za modularne arhitekture implementirane v "Power and Energy Control (PEC)" modulih za upravljanje z močjo in energijo. PEC moduli tako pridobijo modularnost, skalabilnost, visoko dostopnost, povezljivost z drugimi sistemi in nadgradljivost ter kompatibilnost z novejšimi (prihodnjimi) modeli.

1. Introduction

World trends of power consumption are continuously increasing, causing high or even excessive power demands in distribution grids as well as in energy generation. Several initiatives have been started under the term "Demand Response" or "Automatic Demand Response", with the goal to reduce the total or peak power consumption. Some of the available solutions for power and energy management are based for instance on ZigBee wireless devices /1/ or Echelon distributed wired devices /2/, while Siemens /3/ and General Electric provide systems based on Programmable Logic Controllers (PLC's). Entirely distributed systems, such as ZigBee and Echelon, may have problems with availability due to unstable communication channel between control units, which is for instance at 2.4 GHz for ZigBee not guaranteed. Furthermore, distributed solutions exhibit time delay issues when the number of controlled devices is increasing. PLC based systems, on the other hand, are costly and are typically used in the industry, where in addition to power and energy management they are also used to control industrial processes.

Particularly important requirements for power and energy management systems are their scalability and upgradeability. This is calling for a modular approach, which has already been introduced by profibus /4/, modbus /5/, echelon /2/, linbus /6/ and others, providing suitable protocols for communication between modules. Typical weaknesses of those protocols are as follows:

- Protocols are in most cases not self-describing, so a custom application is required in order to be able to understand, monitor and control module's parameters.
- Protocols may introduce significant overhead (such as in the case of widely used Extensible Markup Language, XML), or when compressed, they may only be manageable by powerful devices.
- Master-slave oriented architecture makes modules non-autonomous and dependent.
- Static binding of variables limits future upgradability and compatibility across modules.

In this paper we propose an instrumentation bus with tiny distributed protocol stack. This protocol is a subset of a Sensor Standard General Language (SSGL) /7/, a protocol stack originally developed for low-power sensor networks. We refer to the proposed hardware implementation of the instrumentation bus and the respective protocol stack as Sensor Instrumentation Bus (SIB) and consider modular units attached to the bus as distributed sensors and actuators. We implemented SIB for interconnection of Power and Energy Control (PEC) modules, designed to limit and reduce maximum power demand requirements within households and industry, and to provide energy saving functions. PECs follow modular approach and contain distributed protocol stack with the aim to offer high-availability, redundancy, future upgradeability and installation simplicity.

The rest of the paper is organised as follows. In Section 2 we briefly describe hardware features and modules of SIB

instrumentation bus. In Section 3 we provide an overview of the SIB protocol. Section 4 presents PEC modules as compact autonomous units implementing the proposed SIB instrumentation bus and protocol stack, and Section 5 concludes the paper.

2. Sensor Instrumentation Bus

The main purpose of SIB is to provide low cost instrumentation platform that suits applications in households, industrial fields, automated test environment (ATE) and low-power sensor networks. In the following we provide an overview of its basic features and structure of its modules.

2.1 SIB Basic Features

SIB is based on standard and well established technologies:

- DIN housing suitable for standard electric cases.
- Low-cost communication Inter-Integrated Circuit I2C bus.
- Universal Asynchronous Receiver/Transmitter (UART).
- High-speed point-to-point or multi-point Low-Voltage Differential Signalling (LVDS).

SIB provides decentralised control for a cluster of completely independent modules, where:

- Each module is autonomous and thus self-sufficient.
- Communication between modules is completely decentralised, with no central coordinator nor central processing module.
- Cluster consists of *n* modules stacked in an array sharing a global communication channel and having direct communication channels with adjacent modules to ensure fast communication and short response times.
- Clusters scale via additional isolated communication channel.
- Each module comprehends self-describing protocol for autonomous configuration and understandable human interface.

2.2 SIB Modules

Considering the degree of functionality we define the following classes of modules that can connect via SIB:

- Dependent Modules or Primitive Modules incorporate the minimum set of functions and layers to co-exist in a system and require intelligent modules or bridges for in-system integration. Such modules are typically accessed via I2C or UART bus.
- Independent Modules or Intelligent Modules incorporate all necessary layers to be completely independent. These modules handle communication with other modules on their own. PEC modules belong to this class.
- Bridges provide interfaces to other devices and to external systems for large scale integration. They take control over primitive modules for inter-connection with intelligent modules. Typically a power supply module may integrate such communication bridge.
- Supporting modules may exceptionally come without communication interface at all, such as low-cost power supplies and battery backups.

For its operation a SIB based system requires at least one module belonging to the intelligent class, a power supply, and optionally a bridge providing a connection to PC.

SIB specifies hardware minimalistic communication and power interface for modules, as shown in Figure 1. GLB, LBR and LBL represent communication channels, marked as GLB0, GLB1, or GVL0p/GVL0n when paired in differential mode and using high-speed LVDS. The I2C and GLB0/UART represent the primary communication interface and multi-master access to each device configuration. Fast communication with adjacent modules is achieved via local interfaces LBL and LBR. All signal connections on LBL, LBR and GLB are allowed to pass digital and analogue signals.

POWER up to 24 V and 0 V power signals, up to 3 A
I2C I2C based communication interface
GLB/GVB Global Bus, where pairs can form a LVDS link for very high speed communication
LBR/LVR Local Bus to adjacent module on the right, where pairs can form a LVDS link
LBL/LVL Local Bus to adjacent module on the left, where pairs can form a LVDS link

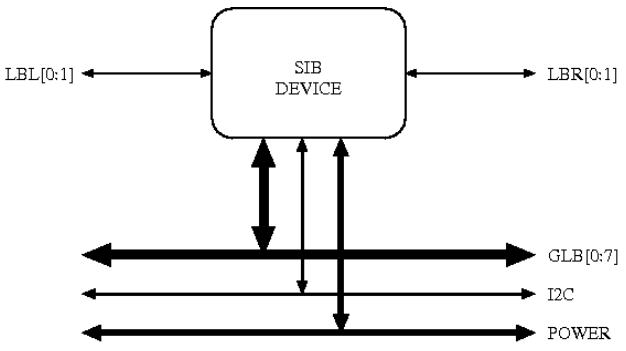

Fig. 1: SIB interface.

3. SIB Protocol Stack

By analogy with Open System Interconnection (OSI) layered architecture /8/ the protocol stack for the proposed instrumentation bus consists of PHY layer (SIB-PHY), medium access layer (SIB-MAC) and message layer (SIB-MSG), the latter encompassing all required functionalities above MAC layer. The proposed layered structure is depicted in Figure 2.

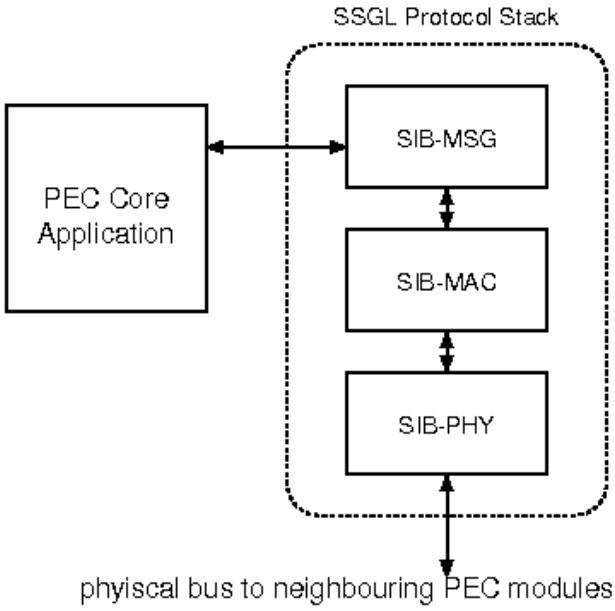

Fig. 2: SIB protocol stack implemented in PEC modules.

3.1 SIB-PHY Layer

SIB-PHY interface implemented in PEC modules only contains five signals that provide connection to left and right modules via two left and right four-pin connectors. These signals include:

- Power signal +24 V to left and right modules.
- Return or GND (0 V) to left and right modules.
- Global communication bus GLB0 to left and right modules.
- Local communication bus LBR0 to right module only.
- Local communication bus LBL0 to left module only.

Communication on the global bus needs to be fail-safe. This means that permanent malfunction of a module should not terminate communication on GLB0. Communication is half-duplex and has multi-master topology where up to 32 devices may be attached to a single GLB0 bus, this limitation being a consequence of power constraints and physical length of the bus.

Physical signalling is based on standard UART interface. With respect to the UART specifications, frame arbitration signalling is added to SIB-PHY, defining basic packet frame or time-slot.

Standard UART low-level framing protocol transfers data on byte basis (similar to linbus /6/) and is configured in 8 bit data mode with odd parity and one stop bit. SIB-PHY adds start of frame and end of frame signals making it suitable for multi-master environments and automatic data rate detection. SIB-PHY UART physical framing is shown in Figure **Error! Reference source not found.** and consist of:

1. Start of Frame (HIGH state) denoting Bus Request. If LOW is detect on the bus prior to step 2, bus is lost, or if HIGH is detected prior asserting logic HIGH, bus is busy. Start of Frame has length of 10 bits (8 HIGH bits + HIGH odd parity bit + one stop bit) and implies the bit rate.
2. Start of Byte (LOW state) begins standard UART transmission of a byte with logic LOW start bit.
3. Individual bits (LOW/HIGH state): the 8 data bits, odd parity.
4. End of Byte (HIGH state): 1 stop bit. *When more bytes are to be sent, repeat from step 2.*
5. End of Frame (LOW state). *When more messages are to be sent, repeat from step 1.*

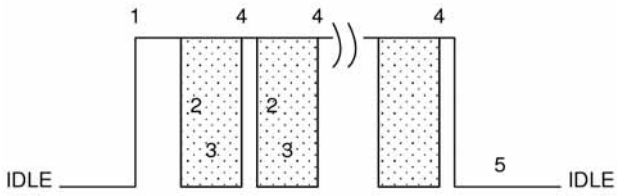

Fig. 3: SIB-PHY UART signalling.

Such physical framing can be implemented by any UART interface found in contemporary microcontrollers (MCUs). The most important properties of such signalling scheme are as follows:

- Transfer is half-duplex, multi-master with frame arbitration and automatic bit-rate detection.
- Bus collision is detected by monitoring (receiving) bytes back from GLB0 during the transmission and also by checking the parity bits.
- Frame length error is detected by standard UART Frame Error Detection mechanism (for instance by BREAK signal, which is generated in this case).
- Typical UARTs in MCUs support a broad selection of bit rates in the range from kbit/s up to Mbit/s.

3.2 SIB-MAC Layer

The SIB-MAC layer defines simple but powerful and low-overhead communication protocol stack. It has been designed to meet requirements of the next generation distributed communications and data processing devices. Being based on SSGL, which was originally developed for sensor networks, its main features include:

- PHY independency at variable data rates.
- Innovative peer-to-peer and broadcast bi-directional pipe.
- Bandwidth allocation with the first order allocation predictor.
- Synchronisation mechanism.
- Acknowledged protocol for transfer reliability.
- Support for low power and power down modes.
- Support for self-management and self-healing of devices.

- Support for redundant systems.
- Ultra-low packet overhead.
- Independency from the device intelligence level.

Full implementation of the SSGL-MAC layer supports standard star and peer-to-peer (full mesh) topologies. It additionally supports a combination of standard peer-to-peer and broadcast modes, simplifying the architecture of redundant and high-availability systems. The last type of connection is named as SSGL-MAC general pipe. For the operation, network does not require any coordinator, making the system very robust.

SSGL-MAC introduces the term pipe on the peer level, which defines transaction between two devices. The SSGL-MAC pipe enables a device to communicate with one peer, forming peer-to-peer connection, and at the same time allowing (or not) others to interrogate into their conversation. More intelligent devices can influence pipe properties, they can recommend better channel and require reduction of the bandwidth. In such a way, pipes can be managed and controlled by other entities without being dependent on them.

SIB only requires the most basic implementation of the SSGL-MAC supporting:

- Totally stripped message header containing only ID and CRC.
- One broadcast pipe and one receive pipe.
- Reception of other responses (messages).
- Internally time-driven broadcast pipe in order to limit maximum bandwidth occupation.
- Data fragmentation to support transfer of long messages.

The SIB-MAC frame consisting of three fields is shown in Figure 4. It can be up to 256 Bytes long. The *Device ID* field (4 Bytes) uniquely defines the device within the array, the *Data* field (0 t 250 Bytes) holds data formatted according to the SIB-MSG layer specification, and the *CRC field* (2 Byte) provides the final checksum.

Preamble	Device ID	Data	CRC	*Ending*
start of frame	4	0-250	2	end of frame

Fig. 4: SIB-MAC frame structure.

SIB devices support only two pipes operating at the same time:

- Broadcast Pipe broadcasts internal information of the device as declared by internal time-slot timer and is capable of receiving responses that are sent by other devices immediately after its transmission. This is the default pipe opened by every device.
- Receive Pipe selectively receives other broadcast messages and responds to the same device. This pipe is opened by the responding device in case there is a message that needs to be transferred only to a specific device with a given ID.

Low level transaction and framing is defined by SIB-PHY layer, however, MAC protocol takes care that sufficient time is left after the broadcast message is sent for safe transmission of responses.

3.3 SIB-MSG Layer

SSGL-MSG layer is designed as a light-weight self-describing protocol so as to suit to simple and advanced devices with minimum data and program memory requirements. Data is carried in the format of a message, which does not require converting the numbers into strings and can carry also structured expressions and powerful equations, including vectors and matrices. Furthermore, such format does not require extensive parsing (such as in the case of XML) by each device as it shifts all the complex arithmetic and logic to the typically more processing powerful devices which are designated to interpret given information. Variable-value pair binding requires only basic parsing, which is manageable also by most simple devices.

The general structure of the SSGL-MSG layer message is shown in Figure 5. It consists of constant (description) and dynamic (arguments) parts. Description is transmitted only upon request, if the message with a given ID is unknown to the receiver. Most of the time only arguments that are changing are (re-)transmitted. Fields marked as *mandatory* are present in every message, while fields marked as *optional* are sent on request or upon a change.

DESCRIPTION	TERMINATOR	ID	ARGUMENTS
optional	*mandatory*	*mandatory*	*optional*
t-bytes	1-byte: 0x00	2-bytes	*a*-bytes

Fig. 5: SSGL message structure.

The constant part of the message is formatted in a similar way as the standard C *printf()* function. It defines how message is to be shown or represented along with its arguments to a human using the following formatting syntax:

- Expressions to support high-level math operations in format *{expr}*.
- Variable-value pair binding with (tolerance) range and multi-dimensional unit support *{:variable}={expr+-tolerance}/unit/*.
- Local and global variables, and cross-referencing: *{:*global}, {:local}*.
- Data structures and unions: *{:struct.substruct.token}, struct{...}*.
- Text formatting, headers, tabular environment, enumerations, lists and buttons.

Each message is an autonomous cell. Multiple messages form larger structures and hence complex contents. Therefore it is required from the MAC to transfer each message as an autonomous cell even if fragmented to smaller pieces.

Each device with SSGL-MSG layer must implement at least two messages:

- INIT MSG - Initialization Message, which has assigned message ID 0. It defines the initial content of a device with unique textual description in human understandable form. This message is typically transferred only once.
- STATUS MSG - Status Message, which has assigned message ID 1. This message describes the current status of a device. In PEC modules, for instance, these include the present power and energy consumption, and the output states.

With respect to SSGL-MSG layer the SIB-MSG layer implementation as required in PEC units is considerably stripped down as it only needs to support the following features for its operation:

- Transmission of constant part of the message.
- Transmission of variable part of the message.
- Handling of local requests for message transmission.
- Handling of remote requests for message transmission.
- Handling of remote requests for message variable update.

The term *local request* refers to operations within MCU, while the term *remote request* refers to all operations sent by other(s) and received via MAC layer.

4. Power and Energy Control Modules

Power and Energy Control modules are compact autonomous units that monitor and control two power electrical branches at a time. They are designed to be placed in a standard (DIN) electric installation housing next to fuses, thus forming a cluster of distributed sensors and actuators. Connection between modules is achieved using contacts placed on the left and the right side of the module 4-pin connector. For the operation they require one (or more for redundancy) power supply unit(s). Typically power supply unit also features a bridge circuit to connect clusters into a larger monitoring and control network. The same interface also provides the access to the user. PEC module as described above is depicted in Figure 6.

A PEC module consists of:

- Isolated current sensor and switch power interface.
- SIB communication interfaces.
- 8-bit Atmel AVR MCU with the SIB protocol stack.
- PEC core application.

In order to prove the concept of SIB instrumentation bus and protocol stack in PEC units we implemented the following devices:

- SIB Bridge Interface that links a PC with PEC units.
- SIB Device Monitor software package in the form of a browser with Contents Viewer capable of high-level presentation of PEC messages and data structures with data input support to write back controllable variables.

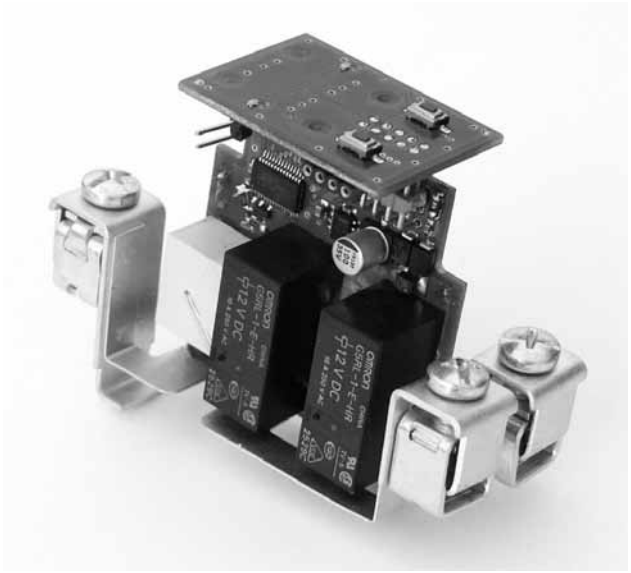

Fig. 6: PEC module implementation.

Description (or constant part of the message) is typically transmitted upon remote requests only until it is known to the receiving device. Description length is typically much longer than the maximum message size, so data is fragmented by SIB-MAC layer into smaller fragments and sent individually. SIB-MAC layer uses fragment counter and start and end of fragment bits to correctly assemble received fragments in the message.

During its operation a PEC unit continuously updates variables in the data structure, which is directly mapped to the variable part of the messages. The application can also issue a local request for (re-)transmission of the message with ID, which has updated variable part of the message. Only variable part of the message is sent with complete message length typically equal to 7 bytes.

By gathering all of the message descriptions along with variables the presentation software builds an HTML page for the PEC device.

In the same way as variables are transmitted from the device, they can be sent back to the device as an input. Using the same ID and message structure, a message is sent using a reply mechanism of the MAC layer (using receive pipe) and received as other response by receiving device. MAC layer passes valid messages to the Message layer and validates input argument(s).

The presentation software uses the same equations that are used in formatting messages into HTML format in reverse direction by computing inverse. Presently supported numerical method is 1-dimensional Newton's method

capable of computing an arbitrary function of a single argument: $x = f^{-1}(y)$.

The graphical user interface of the SIB Device Monitor is shown in Figure 7. It lists all available devices with the first two messages: Initialization (INIT MSG) and Status (STATUS MSG) messages. A mouse click on a device invokes Contents Viewer as depicted in Figure 8, showing:

- Formatted message contents in HTML format on the right side of the figure.
- A list of variables with present values on the left side of the figure. A mouse click on a variable invokes input method for a given type (string, number, list / enumeration, etc.).
- Additional views on the left side of the figure are Message Contents, showing headings sorted in a tree, and Raw Messages that can be used as a debugging tool.

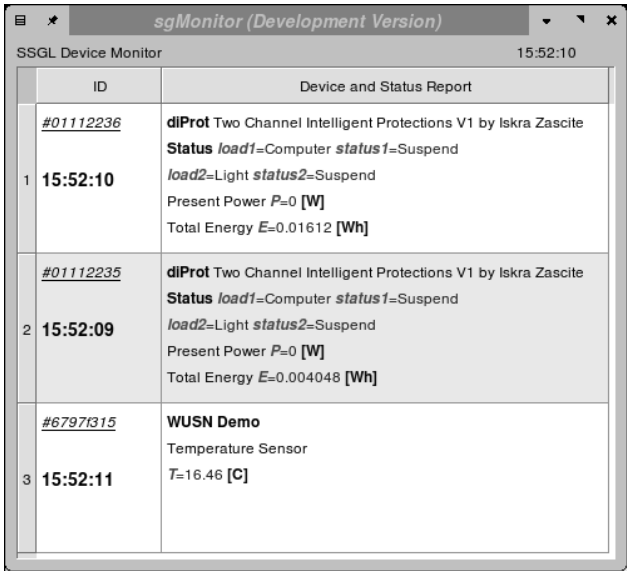

Fig. 7: SSGL Device Monitor.

5. Conclusions and future work

In this paper we described the implementation of the inter-module communication as defined by SIB instrumentation bus and a simplified protocol stack, which was originally defined by SSGL specifications for sensor networks. As a show case we applied the SIB instrumentation bus in power and energy management system consisting of PEC modules. Using the proposed SIB architecture two or more PEC modules can be arranged in an array forming a cluster of instrumentation and control devices with the following major benefits:

- Each PEC module (or SIB module) operates as an autonomous cell providing independency from other modules and consequently higher availability of the entire system compared to centralised architectures.
- SIB protocol stack supports fault-tolerant, multi-master half-duplex communication with collision detection, implemented using standard UART interface and innovative MAC layer broadcast/receiving pipe concepts.
- Being designed as a light-weight protocol it releases CPU from high processing requirements, which is particularly important for low-cost 8-bit devices while operating at high communication speed (Mbit/s).
- Self-describing message formats are featuring rich text formatting and expression post-processing capabilities with (user) feedback input, thus guaranteeing future upgradability and compatibility.
- Presentation software generates standard HTML output, viewable in a browser on any PC, handheld or mobile phone.
- SIB represents a compact solution at low manufacturing costs.

While in this paper demonstrated on PEC modules, the proposed SIB specification can be used for general purpose instrumentation, such as I/O units, power supplies, current and voltage meters, etc. In this respect we are planning to release the SIB specifications for general use and also provide software support library for data acquisition and control.

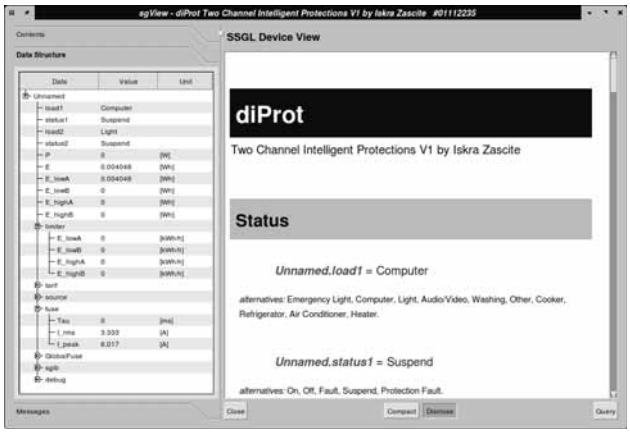

Fig. 8: SSGL Device View.

On the other hand, PEC modules can be integrated in a telemetry and telecontrol system for efficient and reliable remote management of an electric power distribution system, possibly making use of professional wireless communications technology such as TETRA (TErrestrial Trunked RAdio) /9, 10/.

Acknowledgement

The instrumentation bus technology and PEC modules were developed by Iskra Zaščite Ltd. and Insitut "Jožef Stefan".

References

/1/ ZigBee Alliance, http://www.zigbee.org

/2/ Echelon Corporation, http://www.echelon.com

/3/ Siemens Corporation, http:///www.siemens.com

/4/ The Industrial Communications Community, http://www.profibus.com

/5/ Modbus IDA, http://www.modbus.org

/6/ LIN SubBus, http://www.lin-subbus.org/

/7/ Sensor Standard General Language, http://sensornet.ijs.si/ssgl_whitepaper.html

/8/ H. Zimmerman, "OSI Reference Model. The ISO Model of Architecture for Open Systems Connection", IEEE Transactions on Communications, Vol. Com-28, No. 4, April 1980.

/9/ M. Smolnikar, A. Hrovat, M. Mohorčič, I. Ozimek, T. Celcer, G. Kandus, "Telemetry and telecontrol over TETRA network", Inf. MIDEM, 2008, vol. 38, no. 1, pp. 61-68.

/10/ I. Ozimek, T. Javornik, G. Kandus, M. Švajger, "Using TETRA for remote control, supervision and electricity metering in an electric power distribution system", WSEAS trans. commun., 2008, vol. 7, no. 4, pp. 289-299.

Uroš Platiše and Mihael Mohorčič
Institut "Jožef Stefan", Ljubljana, Slovenija

Prispelo (Arrived): 22.07.2009 *Sprejeto (Accepted): 09.03.2010*