The Contribution to Industrial Communication Standards

Juraj Ďuďák*
Institute of Computer systems and Networks
Faculty of Informatics and Information Technologies
Slovak University of Technology in Bratislava
Ilkovičova 3, 842 16 Bratislava, Slovakia
jdudak@gmail.com

Abstract

Description [6, 23, 22, 3] of industrial application protocol MODBUS. The MODBUS protocol principle at lower layers of ISO/OSI reference model. MODBUS over serial line–protocol at second layer of reference model ISO/OSI. State diagrams of communication nodes in MODBUS protocol–master and slave. The weak point analysis of MODBUS over serial line protocol and possibilities of its extension. The propositions of MODBUS protocol extension. Functional specification of proposed modifications. Description of the formal method "Coloured Petri Net", which is used for description of parallel processes and communication protocols. The creation of model communication in MODBUS over serial line protocol used formal method Coloured Petri Nets. The creation of models proposed extensions of MODBUS protocol. The verification of created models by the state space analysis. The implementation of proposed MOBUS protocol modifications titled as mMODBUS on the Cypress microcontroller family.

Categories and Subject Descriptors

D.2.2 [**Design Tools and Techniques**]: Petri nets, State diagrams; C.2.2 [**Network Protocols**]: Protocol—*architecture, verification*; J.7 [**Computers in others systems**]: Industrial control

Keywords

Coloured Petri Nets, MODBUS protocol model, modifications of MODBUS protocol, verification of created model, state space analysis

*Recommended by thesis supervisor:
Assoc. Prof. Pavel Čičák
Defended at Faculty of Informatics and Information Technologies, Slovak University of Technology in Bratislava on April 30, 2010.

Ďuďák, J. The Contribution to Industrial Communication Standards. Information Sciences and Technologies Bulletin of the ACM Slovakia, Vol. 2, No. 1 (2010) 34-41

1. Introduction

At the time, when it is required to interconnect as many as possible technical devices, the communication protocols used to ensure this link are essential. On choosing the appropriate communication protocol depends the overall effectivity of established connections. When choosing the appropriate communication protocol we are limited by the speed requirements and safety, while respecting the defined standards of communication protocols. Apart from Internet/intranet type of networks, networks can be divided into industrial [6], where belongs e.g. Fieldbus, PROFIBUS, PROFINET, CAN, AS-interface, Measurement bus, DeviceNet, M-Bus, industrial ethernet and others [23] and laboratory, e.g. PXI [22] VXI [3] and GPIB [25]. The difference between these categories is that the laboratory measurement system does not expect a great length of the bus as the complex measuring instruments (Oscilloscope, signal generator, multifunction meter, etc.) are connected to the bus terminal nodes and the buses themselves are designed as an extension of I/O computer ports to which they are attached. At the industrial buses is expected greater physical size of the entire system, end nodes are not as complicated as those in laboratory buses and used transfer bus is independent of the internal architecture of the monitoring equipment (control computer). Nowadays, in the area of industrial buses occurs trend to unify these standards and to eliminate their mutual incompatibility. Suitable candidates for the unification of protocols are often simple solutions that use existing standards and add an advanced functionality to them, which allows us to communicate between different communication systems. After a thorough review of the current state of the area of industrial buses, we can seea variety and differences of communication protocols. The unifying element of these standards appears to be open protocol of a higher level that is widely accepted. Such a standard is the Ethernet environment with family of TCP/IP [5] (UDP/IP) protocols. At the selection of protocol, which we will continue to work with, there were raised the following requirements

1. Broadness. The system we will work with falls under the area of local control (from one up to tens of meters).
2. Easy implementing of the communication protocol interface into programmable microcontrollers.
3. Protocol cooperation with higher level protocols (TCP/IP)
4. Ability to extend the functionality of the protocol.

As candidates for the universal use are the protocols MODBUS, CAN, AS-Interface, PROFIBUS and DeviceNet. From these protocols, MODBUS is the most suitable for a number of reasons:

- is suitable for use in the area of local control buses because as a physical layer can be used RS-485 or RS-232,
- the principle of communication (master/slave) and the protocol specification itself (statechart diagrams of master and slave nodes) are easily implementable into current programmable microcontrollers (for example PSoC 1 by Cypress[1])
- natively cooperates with higher level protocols (TCP / IP),
- MODBUS protocol specification is open and easy to modify.

MODBUS Protocol [19], [17] is a protocol that is easy to implement in industrial communication systems and also adaptable to specific applications. MODBUS protocol is suitable to use in systems where the devices are connected to microcontrollers with a certain function, such as. in [14] where the author uses the MODBUS protocol for transmission of measured data, in [9] is MODBUS used in system for electric drives, or [24] where it is used as a communication protocol between the DSP system and a computer (via serial interface port). In [7] the author deals with master node authentication in the MODBUS protocol.

2. Thesis objectives

The main motivation of the thesis is to increase the usability of MODBUS protocol for applications of the local control, which envisages the use of slave devices with some degree of intelligence, respectively logic, adapt the MODBUS protocol to the current industrial systems designed for simplicity and efficiency. The way to achieve this is to identify weak points of MODBUS protocol and propose solutions to address them. The objectives of this work are as follows:

1. Create a MODBUS serial line protocol model using Coloured Petri nets.
2. Propose MODBUS protocol extensions with a goal of wider usability in real industrial applications.
3. Create models of the proposed extensions, verify and implement them in a microprocessor application.

3. Methods and design

For modeling protocols and parallel processes there are several formal techniques. PVS Verification System [1] – a tool for formal specification and verification of hardware, sequential and distributed algorithms. Symbolic Analysis Laboratory [13] (SAL) – a tool for formal specification and verification of communication protocols in particular [4]. Tool PROMELA (Protocol Meta Language) [1], [13] for formal verification of distributed software systems. The system AsmL (Abstract State Machine Language) [21] – a language for formal description of abstract state machines. Markov chains [2] are suitable for modeling of stochastic processes [20]. For the description of communication protocols we have chosen methods that suit best for this purpose: Coloured Petri Nets (CPN) and state machines (Finite State Machine–FSM). State machines are used in description of the functionality of the protocol, Coloured Petri nets are used at creating a functional model of the protocol. In developing the model is based on the state machine, respectively the state diagram for the given protocol.

3.1 Coloured Petri Nets

Colored Petri Nets [11] extend Petri nets [15] on data types, time, and hierarchical structure. They are used mainly for modeling communication protocols and services [10], [16], [12]. Non-hierarchical Coloured Petri Net is n-tuple $CPN = \{\Sigma, P, T, A, N, C, G, E, IN\}$, where single parts have the following meanings:

Σ is a finite set of data types, called colour set. Each colour set must be final and not null.
P is a finite set of *places*.
T is a finite set of *transitions*.
A is a finite set of *edges* such that $P \cap T = P \cap A = T \cap A = \emptyset$
N is a function of a *node*. It is defined from A to $P \times T \cup T \times P$
C is a colouring function. It is defined from P to Σ
G is a *guard* function. It is defined from T to the expression, which satisfies the condition:

$$\forall t \in T : [Type\,(G\,(t)) = Bool \wedge Type\,(Var\,(G\,(t))) \subseteq \Sigma]$$

E is a function of edge expression. It is defined from A to the expression, which satisfies the condition:

$$\forall a \in A :$$

$$\left[Type\,(E\,(a)) = C(p\,(a))_{MS} \wedge Type\,(Var\,(E\,(a))) \subseteq \Sigma\right]$$

where p(a) is a place from the expression N(a).
IN is an initialization function. It is defined from P to the expression, which satisfies the condition:

$$\forall p \in P : \left[Type\,(IN\,(p)) = C(p)_{MS} \wedge Var\,(IN\,(p)) = \emptyset\right]$$

4. Proposition of MODBUS protocol modifications

On closer examination of MODBUS protocol were also found its weaknesses:

- Limited functionality of slave node. The original specification of the MODBUS under slave device understands a simple device with a locked-down functionality.
- Communication is only master/slave exclusively. This type of communication ensures no collisions on the bus. In specific cases we need slave/master communication, while maintaining the characteristics of no collision communication.

In the thesis is proposed expansion and modification of the MODBUS protocol, called mMODBUS. This new specification is based on the original MODBUS serial line protocol specification. The target of the proposed specification mMODBUS is to increase usability of MODBUS protocol in real industrial applications.

[1]http://www.cypress.com/?id=1573

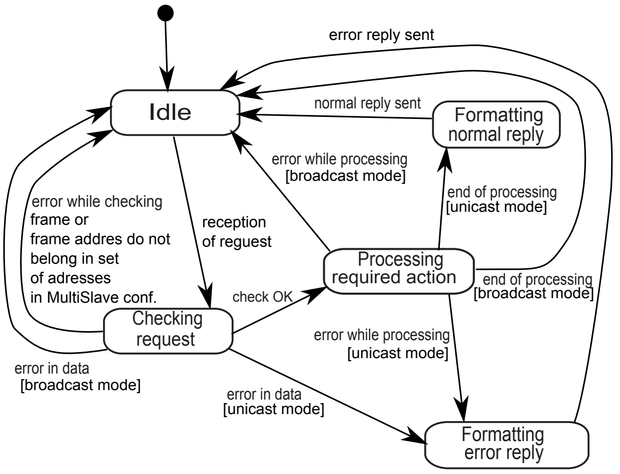

Figure 1: Proposed state diagram for MultiSlave extension

4.1 mMODBUS modification MultiSlave

The motivation for this extension proposal was to use the performance capacity of the devices with implemented slave node. Since the slave node is in many cases software implemented on the programmable microcontrollers, which have disproportionately large computing power in view of the implemented function (display, measuring probe, ...), so the first proposal of mMODBUS protocol is to use the remaining power and to implement several slave nodes on one microcontroller. For the MultiSlave node are proposed following modifications:

1. MultiSlave node consists of a set of slave nodes as defined in the MODBUS specification.
2. MultiSlave node is assigned a set of addresses.
   (a) A message with address that belongs to the set of MultiSlave node addresses, must be accepted by MultiSlave node.
   (b) Number of MultiSlave node addresses is more than 1. The address must be assigned to the MultiSlave node hardware blocks and also virtual hardware blocks (virtual block function is provided by software).
3. MultiSlave node modules are interconnected to the subsystem which can communicate with each other.
4. MultiSlave node responds to data frames, which format was defined in the MODBUS protocol specification. Field *adress* in this response has a value of that address, which was a in the requirement.
5. State diagram of MultiSlave node is in Figure 1.

MultiSlave node state diagram is based on the slave node diagram, as defined in [18]. Proposed state diagram has an enhanced control of the address in the received frame in case of a MultiSlave node type. Received frame is accepted in case where the address belongs to set of the MultiSlave node addresses.

4.2 mMODBUS modification SuperSlave

The motivation for proposing this extension was to allow slave/master communication in specific cases. The need for such communication can occur where there is a need to transfer certain information (such as a real-time information) to all slave nodes, but this information is only available in a one slave device. The proposed modification allows for the adjustment of slave node functionality, where the master device can for precisely defined time delegate the right to send broadcast messages to the specific slave device. Solving the previous situation with this modification is as follows: the master device sends a request to the slave node with a request to send time information to all other slave nodes. Slave device sends a broadcast message with a required data. After a defined time the given the slave device loses its right to transmit data and the master device can handle another requirements. For SuperSlave node are proposed following modification:

1. SuperSlave node must be backward compatible, meaning that it must include the full functionality of the slave node.
2. Upon receipt of the special requirement, SuperSlave node processes the request and sends the processing results to the bus as a broadcast message.
   (a) Right to send the broadcast message to the bus has the SuperSlave node only in a precisely defined time (tw). This time is defined directly in the received message.
   (b) After this time SuperSlave node must not start sending the message.
   (c) SuperSlave node can send at this time only one broadcast message
3. SuperSlave node replies to request for the broadcast transmission only by broadcast message.
4. SuperSlave node can not send unicast messages as a reply to a request for broadcast transmission.

In the original specification of the MODBUS serial line protocol, slave node must respond to each unicast request by a message that contains the address of the sending slave node. Function code in response is the same as in the request (only in case of success). The modification SuperSlave emitting obligation to answer such message, since the slave node would have to reply to master node and then send a broadcast message. However, this solution does not produce the expected improvements in the protocol performance. If the slave node is required to do broadcast transfer, then as a response to the request is to be taken precisely this type of broadcast transmission. The proposal for the SuperSlave node state diagram is in Figure 2. From the original diagram it differs in added state "Waiting for the requested event", to which it gets only if the requirements check went through without problems and it is a super-request. In this state SuperSlave device remains for a maximum period defined in section 2a of the SuperSlave node definition. If the SuperSlave node manages to get the request from the received message done in time, it sends the broadcast message as a reply. In case that by the time tw node SuperSlave does not fulfill the requirement, it must not send any reply. In both cases, the status changes from "Waiting for the desired event" to "Idle". In the proposed modifications are some changes related to the master device. Therefore we suggest to add the following rules for the master device :

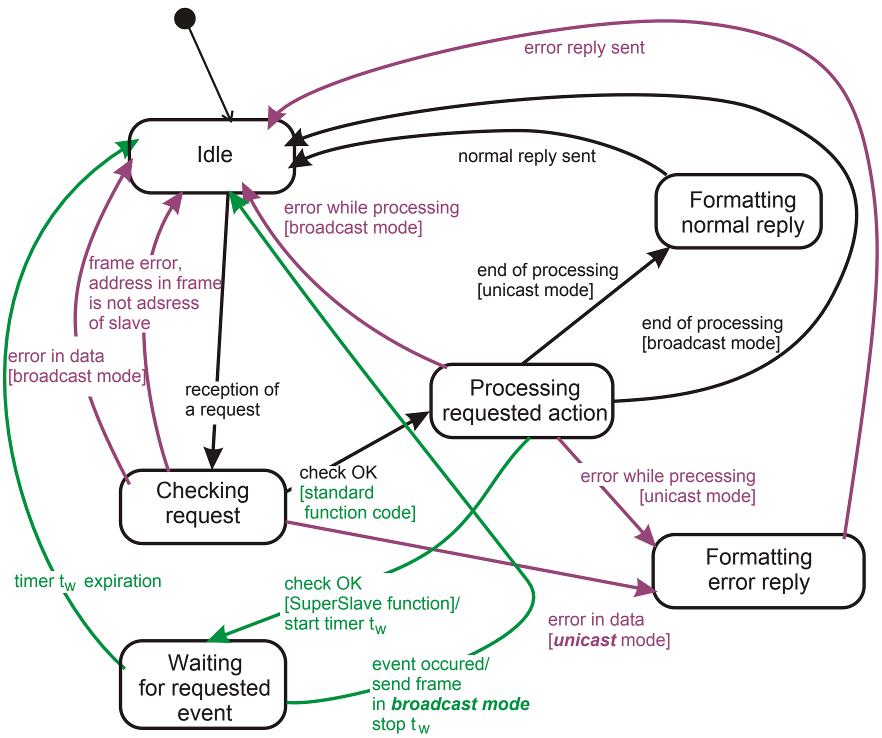

Figure 2: Proposed state diagram for SuperSlave extension

1. Master node must be backward compatible with the specification of the MODBUS serial line protocol.
2. Special requests for SuperSlave node can only be sent as an unicast message.
3. After sending the special request SuperSlave node will wait for the completion of processing for a well-defined time (t3 – relation 1).
   (a) In response to such request the master node must accept only broadcast message.
   (b) In case the t3 time has expired and the answer was not received from SuperSlave node, SuperMaster device will start processing this error condition. According to the nature of the application the request can be re-sent or discarded.

Representation of time t3:

$$t_3 = t_w + t_{BC} + t_2 \qquad (1)$$

In Figure 3 are outlined proposed changes to the master node state diagram. Changes are shown in green (if it is a communication without any error) and red (action of faulty state) color. Where tw is the maximum time allowed for the SuperSlave node to process the request. In the status diagram in Figure 2, this time is represented by the state ”Waiting for the desired event”. Waiting in this state includes requesting that part of SuperSlave node which is to process the request and generate response. It can be a part of the SuperSlave node e.g. real time clock. SuperSlave node receives a super-request, that part of the SuperSlave node, which communicates with real time clock takes over processing. Reading this value and generating the broadcast reply can not take longer than the time tw. Value of time tw may not be constant, so this value is a part of the data in super-request. The value of t3 time is dependent on the running application, because the value of time tw is determined by the master node, respectively by user application according to the nature of super-request. The value of t2 is defined as the waiting time of master device after sending the broadcast request. The value of t_{BC} can be for the given bandwidth f_p derived as:

$$t_{BC} = \frac{frame\,length}{f_p} \cdot 8 \qquad (2)$$

Which is at the 256 bytes of maximum length of data frame and a transfer rate of 9600 bauds, value of approximately 210 ms. Times t_{BC} and t2 are approximately equal, thus the value of time t3 will mainly depend on the value of tw, which is the maximum processing time of super-request, which is specified in the received message. From the nature of applications for which is MODBUS serial line (local control) intended, we opted the maximum value of time tw the same as the maximum value of time t_{BC}. Based on these assumptions we get the maximum value of time t3 equal to circa 620 ms.

5. Model of MODBUS and mMODBUS protocol

In this work is proposed a model of MODBUS and mMODBUS protocol. Creating the model of MODBUS serial line protocol is based on the specification of ”MODBUS over Serial Line Specification and Implementation Guide v1.02 [18]. MODBUS serial line is a second layer of reference model ISO/OSI protocol, thus the model does not include the specification of the first layer, namely the implementation of access methods RTU or ASCII. Model of MODBUS serial line protocol consists of a master node, slave node and a model of transfer media. Models of master and slave nodes are created on the basis of their state diagrams defined in [18].

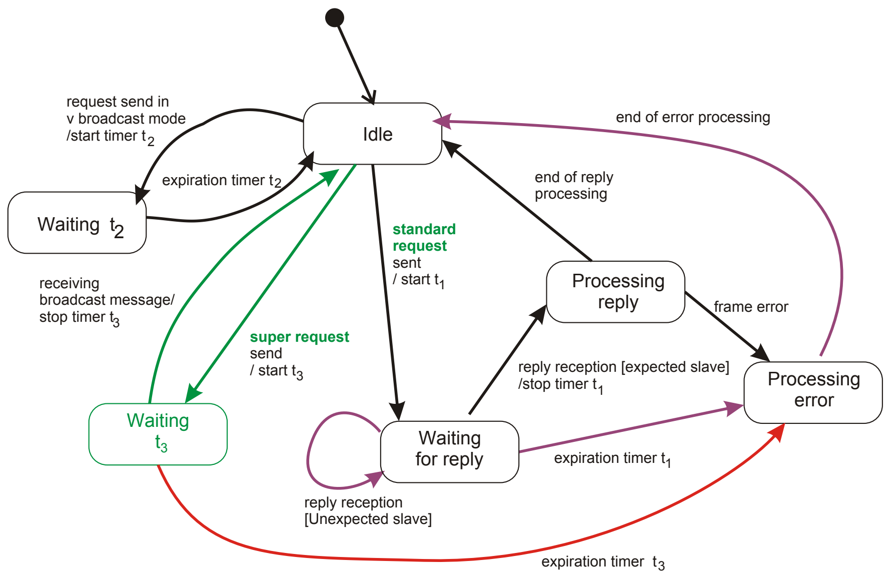

Figure 3: Proposed changes in state diagram for master (SuperSlave extension)

5.1 Model of MODBUS protocol

Model of MODBUS serial line protocol consists of five pages. In addition to these pages, in the model are defined data types, constants, variables and functions which are then used throughout the model. The highest level of the model is MB_hlavna and defines the relation among the master node and the slave nodes. Second level pages are *MB_master*, *MB_slave* and *MODBUS*. Page *MB_master* models the behavior of the master node, *MB_slave* models the behavior of the slave node and *MODBUS* is a model of the communication medium. *MB_hlavna* (Figure 4) is the page of the highest hierarchical level of the model and shows the link between the master node, *MODBUS* bus and slave nodes. The site contains 6 places and 4 substituted transitions. Master subsite is a model of the master node. Places m2b and b2m represent the transfer of the frame from the master node to the bus (m2b – Master To Bus), respectively or transfer of the frame from the bus to the master device (b2m – Bus To Master).
Subpage **MODBUS** models communication channel between master and slave nodes. Places **s2b** (Slave To Bus) and **b2s** (Bus To Slave) model transfer of the frame from slave device to bus and vice versa. In the model in Figure 4 are 2 slave devices connected to the bus – subsite *Slave 1* and *Slave 2*. Each slave node has its configuration stored at the place belonging to the given subpage of the node. Thus, a *Slave 1* node has its configuration stored at the place *conf1* and *Slave 2* node at place *conf2*. In the Figure 4 can be seen that a *Slave 1* node has address 3, the functions codes that is capable of processing are 1, 2, 3, 4 and addresses of its registers, which can be read or written are 1, 2, 3, 4, 5, 6.

6. Verification of proposed models

As a method of verification of the model, we selected the analysis of the state space of CPN network created in modeling software CPNTools [8]. First will be verified MODBUS protocol model. Based on the results of verification of this model, will be verified models of proposed modification mMODBUS.

Choice of verification method

All models of protocols (MODBUS, mMODBUS) are created using Coloured Petri nets in modeling software CPNTools. Techniques to analyze the CPN model are a simulation and calculation of the state space. Simulation of the CPN network provides information on the behavior of the model with random choice of transitions, which can be activated. Analysis of state space provides information about all possible conditions in which the model can be found and all the transitions between these states. For the need of verification of the developed models is an appropriate tools just the state space analysis. Model of MODBUS serial line protocol will be verified as a complex model (the whole model, rather than individual pages) due to mutual functional linkage of the pages of the model.

Definition of the term verification

The term verification means the match of the model behavior with regard to the protocol specification. From the MODBUS protocol specification (chapters 3.2.1 and 3.2.2) result the following arguments:

1. Master node (also the slave) can at one time process only one message.
2. Master node can not send another message until
   (a) did not receive a reply from the slave node (standard or error), has not reached time-out to deliver the message yet – unicast mode,
   (b) has not reached time-out to process the message – broadcast mode.
3. According to the nature of the message sent by the master node the message is delivered
   (a) to one recipient – unicast mode,
   (b) to all slave nodes – broadcast mode.
4. At the communication can not happen the situation (deadlock) which is unable to respond. At the handling of message can not happen infinite loop.

Procedure for verification of the model

State space of model is a graph of occurrence. In the

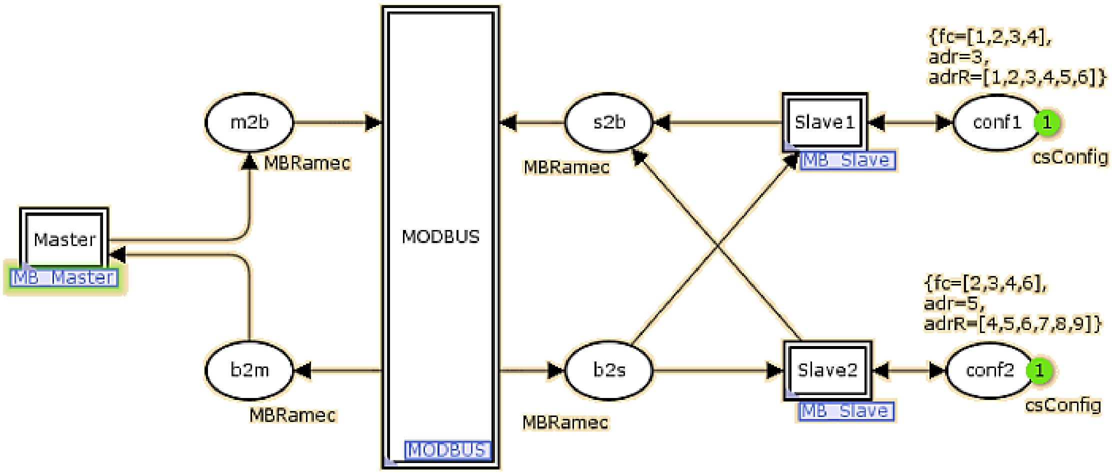

Figure 4: Superpage MB_hlavna

proposed model label *MBRamec* represents sent message. Under the previous assumptions, such a label should appear in the model of master and slave nodes only once (claim 1 and 2). Information on the number of labels anywhere in the model provides the characteristic integer limits, a list of all labels, which existed in the given place is provided by characteristic boundary of multiset. Verification of claim 3 in the analysis of the model means to compare the value of the integer upper limit place **b2s** at CPN page *MB_Hlavna* with the number of connected slave nodes. For a broadcast message, this value must be precisely equal to the value of the number of connected nodes and value equal to 1 for the unicast message. Dead transitions, represent those parts of the model (transitions), not even once activated. Claim 4 can be verified by the number of dead labels, which is the number of final nodes in the graph of occurrence. For the model is expected one dead label. As we are dealing with model analysis, that works with time, more dead labels are expected, which differ only in a time stamp. After removal of time stamp, only one dead label is expected. Condition of no infinite loop existence is satisfied in case, the whole graph of occurrence has been calculated.

6.1 The MODBUS protocol verification

To verify the functionality there are prepared several different scenarios that could occur in communication. Individual scenarios differ in the number and contents of generated messages. At the verification of the functionality of the MODBUS protocol model will be two sets of scenarios – A and B. Given sets of scenarios differ in number of slave nodes that are connected to the bus. In scenario A, there are two slave nodes, and in scenario B are 3-8 connected slave nodes. Individual slave nodes differ in their address (*adr*), a set of code functions (*fc*), which are capable of processing and a set of addresses of their registers (*adrR*). Writing of the slave nodes configuration in the following text is written the same way it is in the subpage *MB_hlavna* (Figure 1).

Scenario A:
Number of connected slaves: 2
Slave nodes configuration:
Slave1: fc=[1,2,3,4], adr=3, adrR=[1,2,3,4,5]
Slave2: fc=[2,3,4,7], adr=5, adrR=[6,7,8,9]

Table 1: Result of state space analysis for scenario A

	adr	codes	PO	SS_n	time [m:s]	PM
A.1	3,5	4,5,7	2	13 905	0:16	7
A.2	0,3,5,6	1,7,20	2	68 101	6:45	24
A.3	0,3,5,6	1,7,20	0	29 245	4:47	6
A.4	0,3,5,6	1,7,20	5	150 957	13:14	50

Scenario B:
Number of connected slaves: 3,5,6,8
Slave nodes configuration:
Slave1: fc=[1,2,3,4], adr=1, adrR=[1,2,3,4,5]
Slave2: fc=[2,3,4,7], adr=2, adrR=[6,7,8,9]
Slave3: fc=[2,3,20,21], adr=3, adrR=[6,7,8,9]
Slave4: fc=[1,2,3,4], adr=4, adrR=[1,2,3,4,5]
Slave5: fc=[1,2,3,4], adr=5, adrR=[1,2,3,4,5]
Slave6: fc=[1,2,3,4], adr=6, adrR=[1,2,3,4,5]
Slave7: fc=[1,2,3,4], adr=7, adrR=[1,2,3,4,5]
Slave8: fc=[1,2,3,4], adr=8, adrR=[1,2,3,4,5]

In Table 1 is described the individual configurations along with the results of the state space analysis. Abbreviations used in Table 1: *adr* and *codes* – addresses and codes of functions used to generate outgoing frames. *PO* – the number of repeated forwarding of ADU frame. *SS_n* – the number of state space nodes. *Time* – the time of calculation of the state space. *DM* – the number of dead labels. For a thorough verification of the functionality of the model, it is needed to connect multiple slave nodes to the bus. MODBUS specification provides for a maximum of 247 slave nodes connected, but in real applications there is a maximum of 16 slave nodes. In this configuration, a maximum of eight slave nodes is connected to the bus. At the attempt to enlarge the model by additional slave nodes, the CPNTools modeling software was not able to calculate the state space model. Scenario B represents a set of MODBUS serial line protocol configurations with one master node and 3-8 slave nodes (in Table 2 it is the column *NS*). In all configurations of scenarios A and B have been ensured match with the assumptions defined in the ”Definition of verification.

Table 2: Result of state space analysis for scenario B

	NS	adr	codes	PO	SS_n	time [m:s]	PM
B.1	3	3,5,6	1,7,20	5	183607	0:19	11
B.2	3	0,3,5, 6	1,7,20	2	170399	0:44	22
B.3	3	0,3,5, 6	1,7,20	2	190167	0:52	24
B.4	5	0,3,5, 6,7	1	1	33893	1:53	3
B.5	6	0,2,3, 4	1	0	295195	24:31	2
B.6	8	1,2,3,4, 5,6,7,8	1	1	68960	0:15	2

Table 3: Configurations of Scenario C

Scenario C	Configuration of slave	
Configuration	Slave 1	Slave 2
C.1	adr=[3]	adr=[5]
C.2	adr=[3]	adr=[5,6]
C.3	adr=[3]	adr=[5,6,7]

6.2 The mMODBUS protocol verification

The model of proposed protocol mMODBUS is based on a model of MODBUS protocol. At the verification of mMODBUS protocol model will be used the same methods and procedures as for the MODBUS protocol.

6.2.1 Verification of extension MultiSlave

To verify the functionality of the MultiSlave node model there are prepared several different configurations, to determine whether all required conditions have been met. Like at the verification of the standard MODBUS serial line protocol model, also here are defined the same assumptions as in the case of MODBUS protocol verification. Verification of the MultiSlave extension model is identified as scenario C. Scenario C will have 3 different configurations of slave nodes:

Scenario C:
Number of connected slaves: 2
Slave nodes configuration:
Slave1: fc=[1,2,3,4], adrR=[1,2,3,4,5]
Slave2: fc=[2,3,4,7], adrR=[6,7,8,9]

In Table 4 is a list of configurations of scenario C with the results of the analysis of the state space.

6.2.2 Verification of extension SuperSlave

For the verification of the SuperSlave extension model we used 2 slave nodes, the first of them was a SuperSlave and the second was a standard slave node. Verification of the model SuperSlave extension model is marked as scenario D. Scenario D will have 4 different configurations (Table 5).

Table 4: Result of state space analysis for scenario C

	adr	codes	PO	SS_n	time [m:s]	PM
C.1	0,3,5,6	1,7,20	2	68014	7:13	24
C.2	0,3,5,6	1,7,20	2	77819	7:56	24
C.3	0,3,5,6, 7,8	1,7,20	1	143440	1:02:53	15

Table 5: Result of state space analysis for scenario D

	adr	codes	PO	SS_n	time [h:m]	PM
D.1	0,3,5,6	1,7,20	2	86546	0:8	24
D.2	3,5	1,7,20,65	2	148520	0:28	28
D.3	0,3,5	1,7,20,65	2	389690	3:52	68
D.4	0,3,5,6	1,7,20,65	1	726624	47:20	31

Scenario D:
Number of connected slaves: 2
Slave nodes configuration:
Slave1: fc=[1,2,3,4,12, **65,66**], adr=3, adrR=[1,2,3,4,5]
Slave2: fc=[2,3,4,7,12], adr=5, adrR=[6,7,8,9]

7. Conclusion

In all configurations of scenarios C and D has been ensured match with the assumptions defined in the part "Definition of verification".

8. Conclusion

In work we proposed mMODBUS protocol specification, based on MODBUS protocol. In this protocol specification is solved more efficient use of current hardware potential of communicating nodes on MODBUS bus, shortening of the response time at the slave-master communication. Based on the draft specification of mMODBUS protocol we created a model of this protocol using the modeling tool Coloured Petri nets. The model was then verified by analyzing the state space model, which examined the behavior of the model in all possible modes of communication. The proposed model was used as the basis for the implementation of the mMODBUS protocol in real industrial application. In the work were reached the following theoretical benefits:

1. Based on MODBUS serial line specification and with regard to the principles of communication in the given protocol, there was designed protocol specification mMODBUS – MultiSlave and SuperSlave. The proposed extensions are fully compatible with standard MODBUS serial line.
2. MODBUS serial line protocol model was created operating at second layer of reference model ISO/OSI using CPN nets. This model can be used as a reference for further modifications.
3. Models of proposed extensions MultiSlave and SuperSlave have been developed using CPN nets.

The functionality of all the developed models was verified by the state space model analysis.

References

[1] Addison-Wesley. *The SPIN MODEL CHECKER - Primer and Reference Manual.* http://spinroot.com/spin/Doc/Book_extras/index.html, isbn 0-321-22862-6 edition, 2007. [Cited: 25. 2. 2010.].

[2] A. T. Bharucha-Reid. *Elements of the Theory of Markov Processes and Their Applications*. Dover Publications, Mineola, N.Y., 1997.

[3] V. B. Consortium. Vmebus extensions for instrumentation. Online, http://www.vxibus.org/files/VXI_Specs/VXI-1_Rev_3.pdf, 2003 december. System Specification VXI-1.

[4] L. de Moura. *Documentation for SAL*. http://sal.csl.sri.com/doc/salenv_tutorial.pdf, 2006. [Cited: 10. 4 2010.].

[5] K. Dostálek, L. *Velký průvodce protokoly TCP/IP a systémem DNS*. Computer Press, 2002. ISBN 80-7226-323-4.

[6] S. Electric. Industrial networks. http://www.automation.schneider-electric.com/as-guide/EN/pdf_files/asg-9-industrial-networks.pdf, september 2009.

[7] L. Gen-Yih. Toward authenticating the master in the modbus protocol. *IEEE TRANSACTIONS ON POWER DELIVERY*, 23(4), 2008.

[8] C. Group. *CPNTools. Computer Tool for Coloured Petri Nets*. University of Aarhus, Denmark, http://wiki.daimi.au.dk/cpntools/_home.wiki, september 2001. [Cited: 22. 8 2009.].

[9] C. Harbin. A novel modbus rtu-based communication system for adjustable speed drives. In *IEEE Vehicle Power and Propulsion Conference (VPPC)*, 978-1-4244-1849-7/08.

[10] G. R. J. Billington, M. Diaz. Application of petri nets to communication networks: Advances in petri nets. *Springer-Verlag*, volume 1605 of Lecture Notes in Computer science, 1999.

[11] K. Jensen. *Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use*, volume Volume 1 Basic Concepts. Springer - Verlag, Berlin, 1997.

[12] K. Jensen. Timed protocol. Technical report, http://wiki.daimi.au.dk/cpntools-help/timed_protocol.wiki?cmd=get&anchor=Timed+Protocol, 2002. [Cited: 22. 8 2009.].

[13] B. Labs. *LTL MODEL CHECKING with SPIN*. http://spinroot.com/spin/whatisspin.html, december 2009. [Cited: 15. 3. 2010.].

[14] C. Lin. Research on voltage acquisition system based on modbus industrial bus and single chip. *IEEE computer society*, International Conference on Signal Processing Systems, 2009. 978-0-7695-3654-5/09.

[15] T. Murata. Petri nets: Properties, analysis and applications. In *Proceedings of the IEEE*, pages 541–580, 1989.

[16] U. of Aarhus. Ring protocol. cpntools - computer tool for coloured petri nets. Technical report, University of Aarhus, http://wiki.daimi.au.dk/cpntools-help/ring_protocol.wiki?cmd=get& anchor=Ring+Protocol, 2002. [Cited: 22. 8 2009.].

[17] M. Organization. *MODBUS Messaging on TCP/IP Implementation Guide V1.0b. Modbus*. http://www.modbus.org/docs/Modbus _Messaging_Implementation_Guide_V1_0b.pdf, october 2006. [Cited: 26. 8 2009.].

[18] M. Organization. Modbus over serial line specification and implementation guide v1.02. modbus. Technical report, http://www.modbus.org/docs/Modbus_over_serial_line_V1_02.pdf, 2006.

[19] M. Organization. *MODBUS PROTOCOL - Modicon Modbus Protocol Reference Guide*. http://www.modbus.org/docs/PI_MBUS_300.pdf, june 2006. [Cited: 26. 8 2009.].

[20] R. Puigjaner. *Performance Modelling of Computer Networks*. ACM New York, La Paz, Bolivia, 2003.

[21] M. Research. *AsmL: Abstract State Machine Language*. Microsoft Corporation, http://research.microsoft.com/en-us/projects/asml/, 2009. [Cited: 15. 3. 2010.].

[22] P. systems Alliance. Pxi specification. Online, http://www.pxisa.org/Spec/pxispec20.pdf, july 2000.

[23] J. Ďuď'ák. Priemyselné a meracie komunikačné štandardy. FIIT STU Bratislava, 2007. Písomná časť dizertačnej skúšky.

[24] Y.-l. Z. Zhi-qiang ZHANG. Realization of communication between dsp and pc based on modbus protocol. In *International Conference on Multimedia Information Networking and Security*, 978-0-7695-3843-3/09, 2009.

[25] N. D. zone. *GPIB Hardware and Software Specifications*. http://zone.ni.com/devzone/cda/tut/p/id/3388, april 2008. [Used: 20. 7 2009.].

Selected Papers by the Author

J. Ďuďák, P. Čičák. CPN MODEL OF THE MODBUS PROTOCOL. In *Mechatronika 2010*, June 3-5,2010, Trenčianske Teplice, in press.

J. Ďuďák, P. Čičák. THE PROPOSITION OF THE FUNCTIONAL EXTENSION OF THE MODBUS PROTOCOL FOR THE INDUSTRIAL APPLICATIONS In *ICSC - Internationl Conference on Soft Computing Applied in Computer and Economic Environments*, (VEGA 1/0649/09), Hodonín: Evropský polytechnický institut, in press

J. Ďuďák, P. Čičák. Model of the Modbus Protocol Using Coloured Petri Nets In *Advanced Simulation of Systems* : Proceedings of the 31st International Autumn Colloquium. - Ostrava: MARQ, 2009. - ISBN 978-80-86840-47-5. - p.121-126, October 20-22, Olomouc (VEGA 1/0649/09)

J. Ďuďák, P. Čičák. Model manažovania spojení v TCP protokole pomocou farebných Petriho sietí In *Komunikačné a informačné technológie*, 5. vedecká konferencia s medzinárodnou účasťou. - Liptovský Mikuláš: AOS, 2009. - ISBN 978-80-8040-376-8. - [6 s.]. Október 14-16,2009, Tatranské Zruby

J. Ďuďák, P. Čičák. Communication Protocols Models by coloured Petri Nets In *Mechatronika 2009*, International conference proceeding, Trenčín, june 3. - 5. 2009. - 200 s. - ISBN 978-80-8075-392-4. - p.109-115.