# Composing of Different Local Area Networks for Industrial Controllers on Common Physical Layer

Georgy Sl. Mihov[1], Emil N. Dimitrov[2], Stoyan N. Jilov[3], Atanas N. Kostadinov[4]

*Abstract* – **In the paper problems of different local area networks have been discussed. Both networks use identical physical layer, but different protocol of exchange. An analysis of existing solutions has been done to establish appropriate hardware and software changes. Distinctive features of each solution have been scheduled. Due to the analysis it is decided to apply the variant of immediate coupling between twin local area networks. Appropriate changes in a protocol for the primary local network have been realized to avoid conflict situations. Also a proper driver for industrial controller to control the secondary local area network has been developed.**

*Keywords* – **local network, industrial controller, network protocol.**

## I. Introduction

One of the main features of Programmable Logical Controllers (PLC) or Industrial Controllers is a capability of networking. Usually each PLC is provided a primary local area network trough witch it is able to exchange information with a same kind of controllers and with a central computer. It is considered that an ability of single local area network supporting is enough for controllers of so called "middle" classes.

However, in many applications, PLC is wanted of supporting secondary local area network in case of control smart sensors and actuators (fig. 1). Usually the secondary network is different kind of primary network. This difference can be appears just in logical level, but it can also be in touch on physical level.

In the paper a method of twin local area networks supporting has been discussed. Both networks use identical physical layer, but different protocol of information exchange. About networks composing appear problems in case that used industrial controller consists of just one serial port trough witch it can be coupled to the networks. The background of the investigation could be scheduled as follow:

1. The used industrial controller is SPV Controller version 4.XX, having just one asynchronous serial communication interface used balanced RS485 standard of EIA.

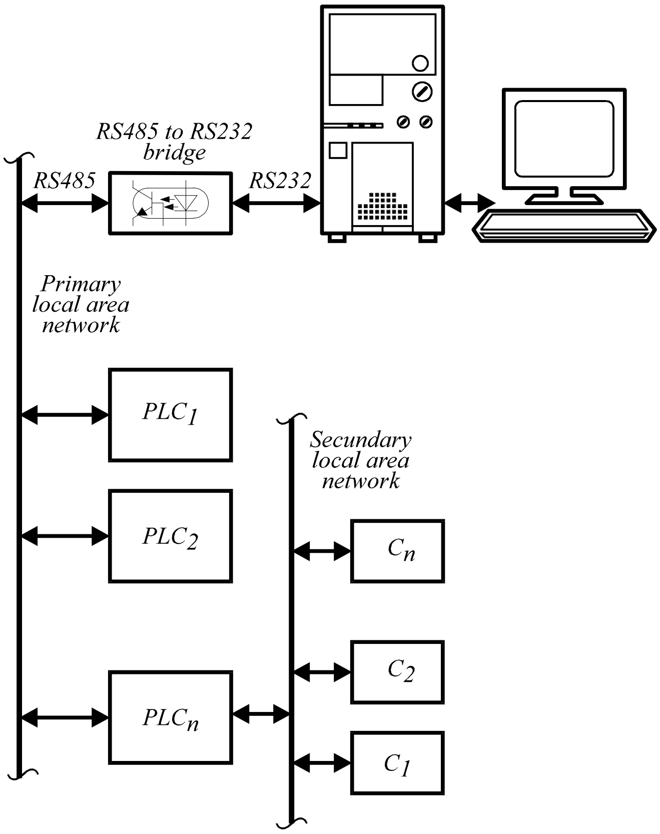


Fig.1. Twin local area network supporting

2. The used industrial controller have to be coupled through the primary local network to controllers of the same rang or to the central computer using SPV protocol for information exchange.

3. The same industrial controller have to be able to control a smart motion controller BERGES ACP 6000 through the secondary local network using serial communication software of Berges electronics for information exchange.

## II. Analysis of Network Conditions

An analysis of existing solutions has been done to establish appropriate hardware and software changes. A universal solution of twin different local area networks composing is shown on fig. 2. It uses a specialized bridge through witch both networks are coupled.

[1]Georgy Sl. Mihov is with the Faculty of Electronic Engineering, & Technologies, TU – Sofia, 1797, Sofia, Bulgaria, E-mail: gsm@vmei.acad.bg
[2]Emil N. Dimitrov is with the Faculty of Electronic Engineering, & Technologies, TU – Sofia, 1797, Sofia, Bulgaria, E-mail: edim@vmei.acad.bg
[3]Stoyan N. Jilov is with the SPV Ltd. Rakovsky 135, 6000, St. Zagora, Bulgaria, E-mail: spv@abv.bg
[4]Atanas N. Kostadinov is with the College "Jon Atanasov", 6000, Plovdiv, Bulgaria, E-mail: nasko@mbox.digsys.bg

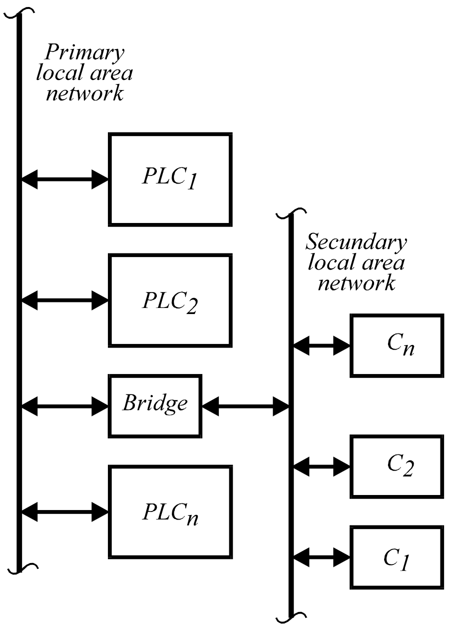


Fig. 2. Using a bridge for two different networks coupling

Distinctive features of this kind of coupling can be scheduled as follow:

1. Primary and secondary local area networks are allowed to use different physical layers. They can use different hardware standard for data exchanging.
2. Primary and secondary local area networks are allowed to use different protocols in logical layers for data exchanging.
3. Each user from primary local area network could establish a connection with a user from secondary network.
4. Users from secondary local area network could appear both as a master and as a slave.

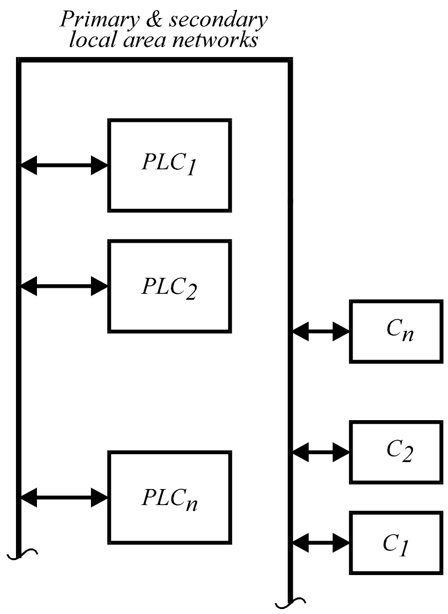


Fig. 3. Immediate coupling between two local area networks

The second solution, that can be used for twin networks coupling, is shown oh fig. 3. It is a direct coupling without any additional devices. The conditions of this kind of connection are firmly limited and could be scheduled:

1. Primary and secondary local area networks must be created on the same physical level and must use the same hardware standard for data exchanging.
2. Different protocols in logical layers for data exchanging may be used, but they have to be evaluated and corrected in case to avoid conflict situations. If it is possible, it is recommended to use the same protocol in primary and secondary networks.
3. Each user from primary local area network could establish a connection with a user from secondary network.
4. Users from secondary local area network could appear as a master and as a slave.

The next solution for twin local area network coupling that has been analyzed is shown on fig. 4. It consists of a line-multiplexing device that switch one user to the primary network or to the secondary network.

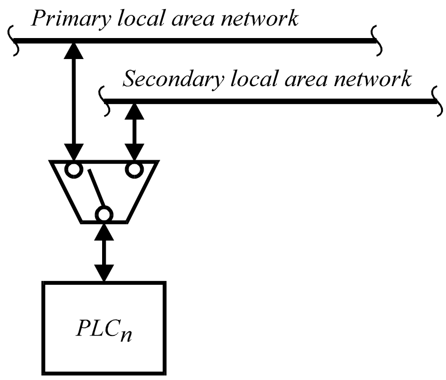


Fig. 4. Using a multiplexing device to connect one user to two different local area networks

The features of this of coupling can be scheduled:

1. Primary and secondary local area networks are allowed to use different hardware standard for data exchanging. The used multiplexing device or has to contain different hardware drivers for each network.
2. Primary and secondary local area networks are allowed to use different protocols in logical layers for data exchanging.
3. Just one user from primary local area network could establish a connection with a user from secondary network.
4. Users from secondary local area network could appear only as a slave.

Due to the analysis of scheduled above kinds of different network couplings it is decided to apply the second variant for mentioned concrete purpose.

## III. Secondary local area network

As it has been written, a serial input output link of BERGES controller has to be used in a secondary local area network. This serial link uses RS485 as the physical standard. The RS485 bus allows 32 differential drivers and 32 differential receivers. It has a noise canceling effect.

The transmitting protocol utilizes two kinds of messages and is made to corresponding to serial input output (SIO) port. The following two fast-select message types are used:

– ENQUIRY message;

– SELECT message.

The SIO link operates on a half-duplex RS485 bus and it is select to operate at speed of 4800 bits per second, utilizing 7 data bits, 1 stop bit, and even parity.

The master uses the ENQUIRY message to request data from one of the slave units and the SELECT telegram for transmission of data to the slave units. There are no provisions for communication between two slaves. Transmission from any slave to the master always needs an ENQUIRY message from the master. Solely the master controls bus access.

When the master sends an ENQUIRY message, the designated slave unit upon receipt will transmit back a reply message according to the transmitted code. ASCII symbol EOT is used to terminate the transmission. The sequence is illustrated on fig. 5.

*Master (enquiry)*

| | | | |
|---|---|---|---|
| *EOT* | *ADR* | *PARAMETER* | *ENQ* |

*Slave (reply)*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| *ADR* | *STX* | *PARAMETER = VALUE* | *ETX* | *BCC* | *or* | *ADR* | *NAK* |

*Master*

| |
|---|
| *EOT* |

Fig. 5. ENQUIRY message transfer structure

Master ENQUIRY message has the following structure:

1. *EOT* – 1 ASCII symbol ‘end of transmission’.
2. *ADR* – 1 character slave address.
3. *PARAMETER* – 5-digit parameter address in ASCII format.
4. *ENQ* – 1 ASCII symbol ‘enquiry’.

After the master sends the ENQUIRY message, the slave sends back a reply message or a negative acknowledge (NAK). Negative acknowledge means that an error occurs during ENQUIRY message receiving.

Slave reply message has the following structure:

1. *ADR* – 1 character slave address.
2. *STX* – 1 ASCII symbol ‘start of text’.
3. *PARAMETER = VALUE* – 5-digit parameter address and 4-digit parameter value, separated with a symbol ‘equal sing’. All data is in ASCII format.
4. *ETX* – 1 ASCII symbol ‘end of text’.
5. BCC – 1 byte block check code, that is an EXOR function of all bytes between STX (excluding) and ETX (including).

The SELECT message permits the transmission of data from the master to the slave units. Upon receipt of the SELECT message, the designated slave will attempt to perform the indicated action, and then send a reply message to the master. EOT is used to terminate the transmission. This sequence is illustrated on fig. 6.

Master select message has the following structure:

1. *EOT* – 1 ASCII symbol ‘end of transmission’.
2. *ADR* – 1 character slave address.
3. *STX* – 1 ASCII symbol ‘start of text’.
4. *PARAMETER = VALUE* – 5-digit parameter address and 4-digit parameter value, separated with a symbol ‘equal sing’. All data is in ASCII format.
5. *ETX* – 1 ASCII symbol ‘end of text’.
6. BCC – 1 byte block check code, that is an EXOR function of all bytes between STX (excluding) and ETX (including).

*Master (select)*

| | | | | | |
|---|---|---|---|---|---|
| *EOT* | *ADR* | *STX* | *PARAMETER = VALUE* | *ETX* | *BCC* |

*Slave (reply)*

| | | | | |
|---|---|---|---|---|
| *ADR* | *ACK* | *or* | *ADR* | *NAK* |

*Master*

| |
|---|
| *EOT* |

Fig. 6. SELECT message transfer structure

After the master sends the select message, the slave send back a positive or negative acknowledge. Positive acknowledge (ACK) means that a valid message is received. Negative acknowledge (NAK) means that a message with a false block check code or invalid parameter code is sent, or access to the control and parameter block is not authorized, or the sending value is outside the range.

## IV. PRIMARY LOKAL AREA NETWORK

A developed in Technical University – Sofia local area network [2] has been used as a primary network. Some additional changes have been done to adapt physical and logical layers. From hardware point of view the physical layer was changed to a standard RS485.

The serial port of the industrial controller is used for data transmission. The data is transmitted asynchronous, in 8 bits format, even control and 1 stop bit. The speed of transmission is chosen to 4800 bits per second. A voltaic split of the local area network is organized. Each receiver and transmitter in network can be supplied realized by common power supply using additional cables. They also could be supplied by own voltaic insulated voltage.

*Message*

| | | | | | | |
|---|---|---|---|---|---|---|
| *SOH* | *RCV* | *TRN* | *LENGTH* | *DATA* | *CKSUM* | *EOT* |

*Reply*

| | | |
|---|---|---|
| *ACK* | *or* | *NAK* |

Fig. 7. Block message structure of the primary local area network

From a software point of view in a protocol for information exchanging have been done some changes. Firstly, the data is transferring with a single block by ASCII format. Due to this, no color byte is needed to identify consecutive block of message. The next change is done in connection with using an ASCII format of information. No more is needed to change the parity control when system or data information is transmitting.

The data packet of the local area network has the following structure (fig. 7):

1. *SOH* – 1 ASCII symbol 'start of header'. This is a leading symbol that indicates the start of the data block transmission.
2. *RCV* – 1 ASCII symbol. It indicates the address of the message's receiver (destination address). It is recommended this symbol to be a letter from A to Z. Usually letter A is reserved for a main station, letter B – for a service station, letter C – for a factory default setting address of an industrial controller and letter Z – for a calibration controller.
3. *TRN* – 1 ASCII symbol. It indicates the address of the message's transmitter (source address).
4. *LENGTH* – 2 ASCII symbols. It indicates block length e.g. the number of the bytes in the informative part.
5. *DATA* – This is an Informative part and can be sized from 0 to 255 ($FF) ASCII bytes.
6. *CKSUM* – 2 ASCII symbols for check sum. It is the least byte of the sum of the all transmitted bytes – from SOH to end of DATA. It is used to check the block receive accuracy.
7. *EOT* – 1 ASCII symbol 'end of transmission'.

After the transmitter sends the message, the receiver sends back a positive or negative acknowledge. Positive acknowledge (ACK) means that the transmission had finished successfully and the calculated checksum equals to the received checksum. Negative acknowledge (NAK) means that the transmitter must resend this message, because an error occurs.

Sending a BREAK signal does the transmission interruption. All users, including the sender recognize the BREAK signal like an error in the format of the transmission data.

All users are listening to the line without interruption and start transmitting a message only when the line is not occupied. The time for one symbol's transmission is limited. The users check for the busy line after the time for one symbol's transmission.

Each sender listens to the line (receives its own transmission) and compares sent and received information. In case of difference (someone transmits over the line too) the sender interrupts the transmission and sends a BREAK signal to inform the others users for conflict. After that each user continues to receive, but does not start transmitting before waiting a time, that depends on user's address in the network. So the conflict situations in the primary network are solved.

## V. Driver for secondary network

In the paper a proper driver for a industrial controller SPV 4.XX to control the secondary local area network has been developed. It realizes all sending and receiving functions, protected times, repeating when an error occurs e.t.c.

The program consists of initialization part, receiver and transmitter. It is similar to the primary network's structure. The driver's entry points are listen below:

1. *Network initialization*. This entry point is used for initialization at the beginning of networking.
2. *Interrupt request from a timer*. This is the driver's 'sense of time'. In this entry point just a modification of the counters of the protected times is done and the processor does not delay.
3. *Interrupt request from serial port*. This is the interrupt from the serial port, that submits interrupt request when a symbol is received form the network and when a transmission of a symbol is done.
4. *Start of communication*. The industrial controller that is coupled to the bought network can be asked only as a master for the secondary network. So, this entry point is activated to confirm all information transfer for communication with slave units from secondary network.

## VI. Conclusions

In the paper a method of twin local area networks supporting has been discussed. Both networks use identical physical layer, but different protocol of exchange. The used industrial controller (SPV Controller version 4.XX) has just one asynchronous serial communication interface based on balanced RS485 standard of EIA. This industrial controller have to be coupled through the primary local network to controllers of the same rang or to the central computer using SPV protocol for information exchange. The same industrial controller have to be able to control a smart motion controller BERGES ACP 6000 through the secondary local network using serial communication software of BERGES electronics for information exchange.

Problems about using one serial port for twin networks are solved. An analysis of existing solutions has been done to establish appropriate hardware and software changes. Distinctive features of each solution have been scheduled. Due to the analysis it is decided to apply the variant with an immediate coupling between twin local area networks.

Appropriate changes in a protocol for the primary local network have been realized to avoid conflict situations. Also a proper driver for industrial controller to control the secondary local area network has been developed.

## References


[1] Mihov G., I. Tashev. Industrial Controller For Discrette Manifacture. National Scientific Conference "Electronics '96", book I, 31-36, Sozopol, Bulgaria, September, 1996.

[2] Dimitrov I., M. Ivanov. Software and Hardware for a Communication Environment in an Information Controlling Systems for Discrete Manufacturing. The 9th International Conference "Electronics '2000", September, Sozopol, Bulgaria.

[3] Dimitrov E., G. Mihov, I. Tashev, M. Mitev. Local Area Network for Industrial Controller. The International Scientific Conference ENERGY AND INFORMATION SYSTEMS AND TECHNOLOGIES. vol. III, pp. 608-613. Bitola, Macedonia, June 7-8, 2001.

[4] ACP 6000 Serial Comunication Software – Manual. Berges electronic. Marienheide. Germany. 1993.