Approaches to Improve of Network Driver for Industrial Purposes

Stanimir D. Mollov[1]

Abstract – **The paper presents the approaches to improve of network driver for industrial purposes. The transmission of the messages over the industrial network has been discussed. An analysis of the serial asynchronous interface has been done. Due to the analyses it is decided to apply the variant of realized an additional special signal, which enables output buffer of the network driver. Possibilities for existing conflict situations have been analyzed. The variants for their overcoming are offered.**

Keywords – **network driver, output buffer, serial interface**

I. Introduction

The driver for the local array network for industrial purposes realizes all sending and receiving functions, protected times, repeating when an error occurs. Besides its own work area in the operating memory, the driver supports and state's word. The driver communicates with the software through that word by setting and cleaning flags. It is consist of: initializing part, receiver and transmitter.

The driver for the local array network has the following entry points shown on fig.1.

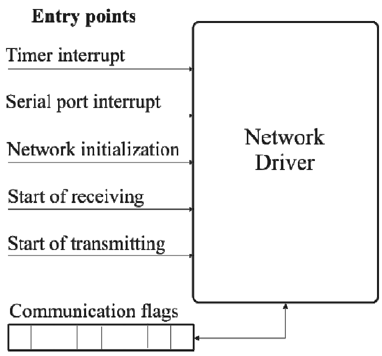

Fig. 1 Entry points of the network driver

Interrupt from system timer – the control is given here in case of interrupt from system timer. This entry point only modifies the counters of the protected times and checks whether the protected time is past.

Interrupt from serial port – this is the interrupt from serial port, which submits request when a symbol is received from the network and when a transmission of a symbol is finished.

[1]Stanimir D. Mollov is with the Faculty of Electronic Engineering, and Technologies, TU – Sofia, 1797, Sofia, Bulgaria, E-mail: smollov@abv.bg

Network initialization – this entry point is being called for initial initialization of the serial port. The receiver and the transmitter are set in inactive state.

Start of receiving – this entry point sets the address of the input buffer and its maximum length. The driver does the necessary changes in the state's word. By this way the driver sets the receiver in active mode. After that the receiver starts to watch the line and to receive the messages for this station. After a message is receiver the receiver marks that by setting the appropriate flags in the state's word and goes into end of receiving mode.

Start of transmission – this entry point is activated only in case of inactive transmitter (the transmission of the last message is finished). Here the address of the output buffer, the length of the message and the recipient are writing. The driver makes the necessary changes in the state's word and sets the transmitter in active mode. From here on the transmitter checks whether the line is busy and when it is not used starts the transmission of the message. After the transmitting had finished it mark by setting the appropriate flags in the state's word and goes into inactive mode.

II. Transmitting the Messages via the Industrial Network

The network driver processes each received symbol. In the receiving mode it watches the line and receives the messages, which are destined for this station. In the transmission mode the driver check whether the line is busy and when is not used starts the transmission of the message.

Due to the high noise level in the communication environment, it is necessary the transferred data to be blocked in comparatively short blocks.

After the sender send the data block via industrial network, the receiver acknowledges its receipt. Positive receipt (ACK) means that the transmission had finished successfully. Negative receipt (NAK) means that the sender must resend this block, because of a block receive error.

All actions in the local area network have own protected times. If the station transmitter had not received the acknowledgment at a fixed time interval, it repeats the transmission of the current data block. If the acknowledgment from the receiving station was positive, but it had not received, due to the noises in the network, the transmitting station has to repeat the same message. The receiver will find out that the block transmission sequence is broken thanks to the block identifier. The transmission interruption is done by sending a BREAK signal. That is recognized like error in the format of transmission data by all station including sender.

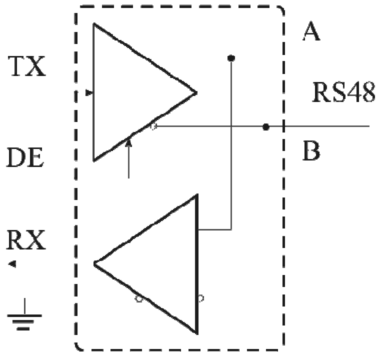

Fig. 2 Output buffer of the network driver

A conflict situation is possible to appear when more than one station from local area network starts transmission at the same time. Each sender listens to the line (receives its own transmission) and compares the sent and the received information. In case of a difference (someone transmits over the line too) the sender interrupts the transmission and sends a BREAK signal to inform the others users for the conflict situation. After that each station continues to receive, but does not start transmission before a certain time, that depends on user's address in the network.

III. Interface for Data Transmission

For bi-directional data transmission usually are used two types of interfaces: half-duplex and full-duplex. The full-duplex interfaces used separated communication lines (fig.3), while half-duplex interfaces used common communication line (fig.4) as for the transmission as well as for the receiving.

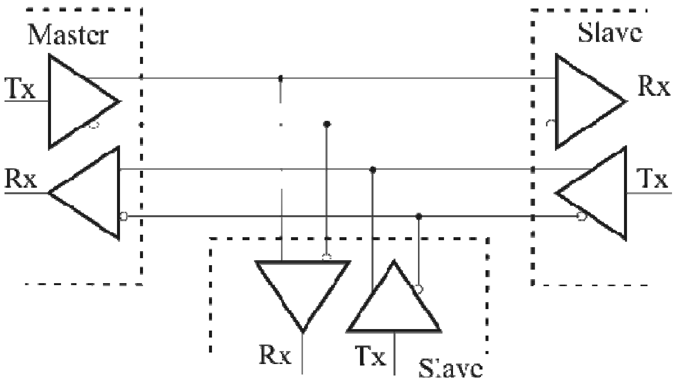

Fig.3 Full –duplex interface

Normally RS485 network uses "half-duplex" interface. All device transmitters and receivers in this network are connected to the same bus. Due to this, it is necessary certain precautions to be taken to avoid the activation of more than one transmitter at the same time. For this purpose a special signal must be created, which enables output driver of transmitter, when some data is transmitting. The correctly determining the position and duration of time interval, during which an output driver on RS485 transmitter is enabled, increases the reliability of information.

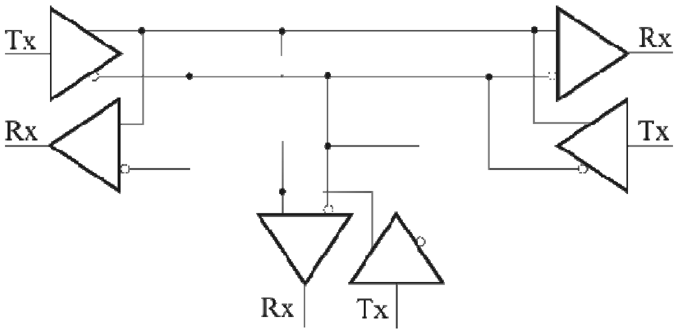

Fig. 4 Half-duplex interface

IV. Serial Asynchronous Interface

The serial communication is a suitable way for connection between controllers or between controllers and other smart devices. It offers a physical interface with a minimum number of lines. Usually the connection between controllers over the industrial network is based on serial communication asynchronous interface. The asynchronous transmission do not supports the clock frequency between transmitter and receiver. Because of this the number of communication line is decreased. Normally the correspondents are clocked by own clock generators. The synchronization between them is accomplished from additional bits, included into the transmitting information.

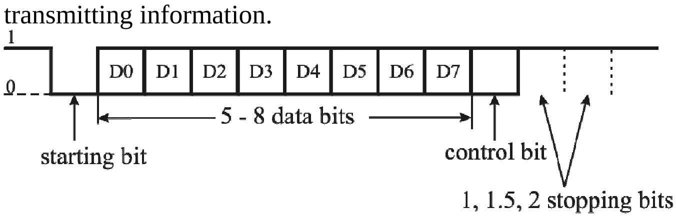

Fig.5 Serial asynchronous standard

The serial asynchronous standard is shown on fig.5. When the line is not active, its state is logical 1. In the beginning the starting bit is sent with active level is logical 0. Immediately follow the data bits. The data byte is transmitted with the least significant bit in front. After that the control bit (parity/no parity) and the 1-2 stopping bits are transmitted.

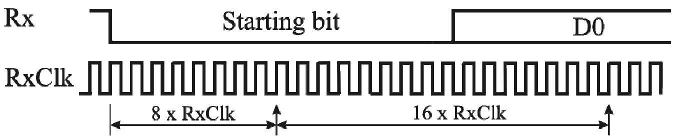

Fig.6 Synchronization between transmitter and receiver

The transmitter and receiver are functionally independent, but use the same data format and baud rate. The baud rate generator fig.7 produces a clock either x16 or x64 of the bit shift rate.

After the first falling edge on Rx line the receiver waiting 8 clocks cycles RxClk and checks the input signal level. If it is logical 1 the next receiving is interrupted. If it is logical 0 the receiver accepts that the starting bit is arrived. After that the remaining bits are receiving at time interval equal to 16 clock cycles of RxClk. By this way all bits are sampled in the middle of its transmission time interval. The changes of the transmission speed may reflect over correctly reception.

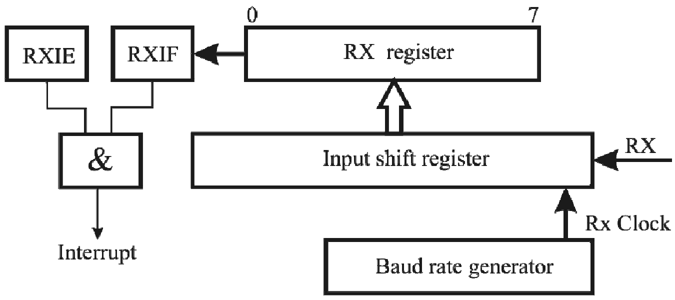

Fig. 7 Asynchronous receiver

The asynchronous receiver block diagram is shown on fig.7. The heart of the receiver is an input shift register. The RX register obtains its data from the shift register. Once the shift register transfers the data to the RX register, the flag bit RXIF is setting. The interrupt can be enabled/disabled by setting/clearing enable bit RXIE. The flag RXIF will reset only when the data is reading from RX register.

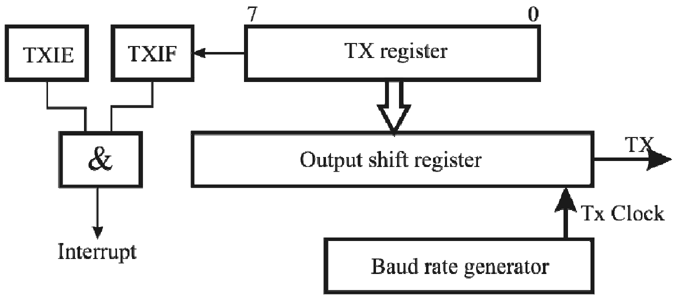

Fig.8 Asynchronous transmitter

The asynchronous transmitter block diagram is shown on fig.8. The shift register obtains its data from the TX register. The TX register is software loading with data. Once the TX register transfers the data to the shift register, the TX register is emptying and flag bit TXIF is setting. The interrupt can be enabled/disabled by setting/clearing enable bit TXIE. The flag TXIF will reset only when new data is loading into the TXREG register.

V. NETWORK DRIVER

A. Transmission Mode

In the transmission mode the network driver checks whether the line is busy or not. If the line is free the driver enable own output buffer (fig.2) and start the transmission of the message. The activating of the output buffer must occur before transmitting starting bit of the first byte of the message. The deactivating of the output buffer must occur after transmitting stopping bit of the last byte of the message. A conflict situation may occur when the transmitter remaining in active state when the answer of the previous message is transmitted via network. In this case the transmitters of two stations are enabled at same time. The transmitter of the first station holds the line in logical 1 (last stopping bit of the message). The transmitter of the second station tries to hold the line in logical 0 (first starting bit of the answer). To get out of this conflict it is necessary to decrease the time for deactivating the output buffer after transmitting the last stopping bit of the message.

Two variant may be used to deactivate output buffer after last stopping bit:

- via interrupt;
- via polling the interrupt flag or another special bits; controlled by output shift register.

To use the first variant it is necessary the interrupt from the transmitter to be enabled. This performs by setting the TXIE bit. When the output shift register is loading from the TX register, the flag bit TXIF is setting and the micro-controller enters into interrupt mode. The deactivating of the output buffer should occur with time delay after setting TXIF. That time interval is equal to the time for physical transmitting of the contents of the output shift register.

It is possible to use interrupt from the receiver if the station listens and compare for a right transmission. After transferring the last byte of the message from TX register into the output shift register the interrupt from receiver is enabling. This interrupt deactivates output driver of the transmitter.

The advantage of using interrupt is that the processor time is used effectively for other tasks. The disadvantage of using interrupt is the necessity of time for storing and restoring the contents of the processor registers when the interrupt is generated.

A special bit, which shows the status of the output shift register, may be used to deactivate of the output buffer. A good example is an asynchronous transmitter embedded in microcontrollers PIC produced by Microchip (fig.9). While TXIF indicated the status of the TXREG register, another bit TRMT shows the status of the TSR register. Status bit TRMT is a read only bit, which sets when the TSR register becomes empty. No interrupt logic is tied to this bit, so it has to be permanently polling to determine if the TSR register is empty. When the TMRT bit is setting, a time delay equal to the time for one bit transmitting must be generated. After this delay the output buffer must be disabled.

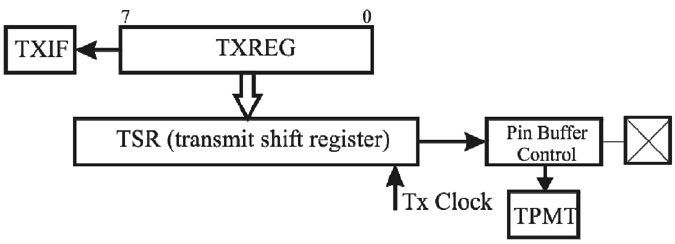

Fig.10 USART transmitter block diagram

B. Receiving Mode

After receiving of the last byte of the message the network driver analyze it. The analysis is accomplished on the base of checksum attached to the end of the message. The station transmitter calculates and attaches the checksum at the end of message. The station receiver also calculates such checksum from incoming bytes of the message. At the end station receiver compares the received and calculated checksum and creates the answer to station transmitter. If these two values are equal the answer is positive else the answer is negative and the transmitter station must repeat the message.

The time for generating the answer of the receiving message may be rather short. In this case it is possible the station receiver to reply before the output buffer of the station transmitter is not disabled (fig.11).

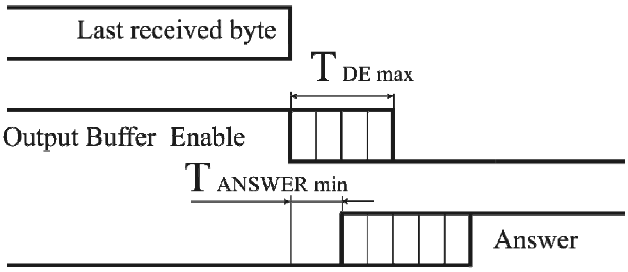

Fig.11 A possibilities for a conflict situation

$T_{DE\ max}$ – maximum time interval for deactivating of the output buffer of the transmitter after the last byte

$T_{ANSWER\ min}$ – minimum time interval for transmitting the answer after receiving the last byte of the message

To avoid from this conflict situation it is necessary the following condition must be keep:

$$\frac{T_{DE\max}}{T_{ANSWER\min}} < \frac{1}{2} T_{Bit}$$

,where T_{Bit} is the time for transmitting a bit. T_{Bit} depends of the used transmission speed.

If this condition is broken the right transmission is not guaranteed.

There are several solutions to avoid from situations:

- using a software delay (all other interrupts may be enable);
- using a timer interrupt;
- using an interrupt from transmitter.

The last one solution is shown on fig .12. The station receiver sends two preliminary symbols before answer. When the preliminary symbol is transmitted the output buffer is not enabled. After loading the preliminary byte into the output shift register the interrupt is generating.

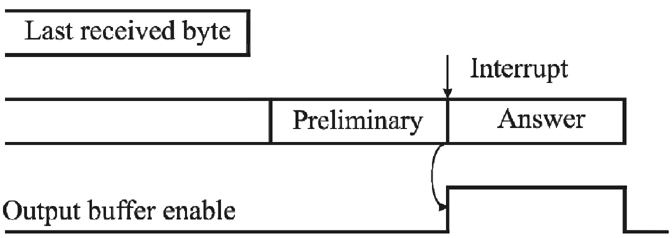

Fig. 12 Interrupt from transmitter

The interrupt loading the answer into TX register and activates the output buffer. In this case a total time delay is equal to time for transmitting the preliminary symbol.

C. Input Buffer

To increase the effectiveness of data exchange speed the input buffer of the network driver is realized as shown in fig.13. After receiving the current message the network driver change the current input buffer address (If been IBUF1 then IBUF2 is a new input buffer and vice versa). So, a new message is allowed to arrive before the previous message is transferred input buffer to input message queue.

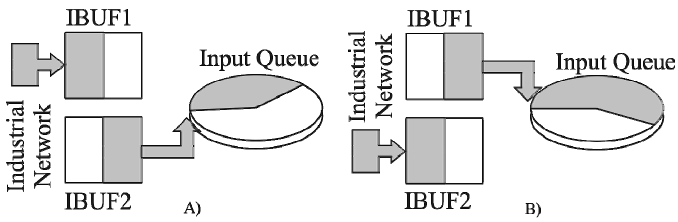

Fig. 13 Input buffer organizing

VI. CONCLUSION

In this paper the approaches for improve of network driver for industrial purposes are discussed. The transmission of the messages over the industrial network is discussed. An analysis of the serial asynchronous interface has been done.

Possibilities for existing conflict situations have been analyzed. The variants for their overcoming are offered.

The article describes also another variant of the enable signal ending, using a special bit which is dependent of the used micro-controller.

In the paper the condition to avoid from conflict situations is present. If this condition is broken the right transmission is not guaranteed.

REFERENCES

[1] Dimitrov, E., G. Mihov, I. Tashev, M. Mitev, *Local array network for industrial controller,* Proceedings of the Int. Scientific Conference ENERGY AND INFORMATION SYSTEM AND TEHNOLOGIES, . vol. 3 pp. 608-613, June 7-8, 2001, Bitola, Macedonia.

[2] Herbert, F. *Implementing Network Protocols and Drivers – with STREAM.* Embedded Systems Programming, April, 1997, p.28.

[3] Herhandez, E., Chidester, and A. George, "*Adaptive Sampling for Network Menagement*", Journal of Network and System Management, Journal Publications from the HCS Research Lab, 1999.

[4] DS41124C – PIC16C745/765 Datasheet, Microchip Technology Inc., 1999