

Operating algorithms for the intelligent control controller of a sucker rod pumping unit

Mahammad Rezvan

Institute of Mathematics, Baku, Azerbaijan

ARTICLE INFO

Article history:

Received 18.11.2025

Received in revised form 04.12.2025

Accepted 12.12.2025

Available online 20.03.2026

Keywords:

Well controller

Analog-to-Digital Converter

MODBUS protocol

Microprocessor

Cyclic Redundancy Check

Dynamometer chart

Radio communication

ABSTRACT

The article examines the operating algorithms of modernized intelligent monitoring and control well controllers (IWC) located at oil wells operated by SRPU, using the capabilities of modern information technology science and new microprocessors. A general block diagram of the modernized IWC is provided, and the MODBUS protocol used during information exchange between the IWC and external devices is formed according to the purpose of each request. The algorithms for the initial parameter determination service of the given block diagram, the units checking for the presence of a request from external devices to the controller, checking the request validity, and the request service blocks are provided, and the operating mechanism of these algorithms is explained extensively.

1. Introduction

It is known that in the final stages of oil field operation, oil extraction from wells is mainly carried out by mechanized methods through high energy-consuming sucker rod pumping units (SRPU), electric submersible pumps (ESP), screw pumps, and other pumps [1]. 41% of wells in the Russian Federation use SRPU, and 54% use ESP. Published data show that more than 85% of mechanized production wells in the USA are equipped with SRPU. SRPU is very popular due to its simplicity, reliability, and wide application possibilities. Currently, despite the creation of other pump types, more than 2/3 of the operating well stock is equipped with SRPU, and their number is steadily increasing. This oil production method will remain the most common for a long time to come [2].

Therefore, the creation of intelligent monitoring and control algorithms for the SRPU oil production method using the capabilities of modern information technology science and new microprocessors retains its relevance. The application of modern microprocessors in distributed control systems allows for a number of measurement and control tasks to be executed directly at the object, making it possible to save on unnecessary information exchanges and increase the overall reliability of the system's operation [3].

Taking all these possibilities into account, the article explores the development of algorithms and software for intelligent monitoring and control controllers (IWC) for oil wells operated by SRPU.

E-mail address: rezvanmahammad@gmail.com (M.H. Rezvan)

2. Problem statement

The intelligent well controller (IWC) is designed to perform the following functions:

- Reading dynamometer chart data from the output signals of stroke and force transmitters (sensors) based on a request from a computer and sending them to the computer via radio communication;
- Automatically reading state signal (SS) data and storing it in the memory of the IWC;
- Executing telecontrol (TC) commands based on a request from a computer and monitoring the execution of these commands;
- Determining the sampling interval by means of a corresponding algorithm by measuring with ultra-high frequency technologies in order to capture useful noise present in the output signals of force and angular displacement transducers [4];
- Calculating position-binary indicators based on the output signals of force and angular displacement transducers for early diagnosis of the SRPU [5];
- Calculating the informative attributes of signals (average pulse frequency, ratio of average pulse lengths to the average value of the pause period, pulse duty factor) by separating the output signals of force and angular displacement transducers into position-binary components in order to ensure control over the onset and development dynamics of SRPU malfunctions [6];
- Ensuring the reading of data from a specific memory address of the IWC and sending it to the computer, or changing the content of a specific memory address based on a request from the computer;
- Compiling and interpreting the general block diagram and the algorithms of its individual blocks for the monitoring, measurement, control, and diagnostic controller, which forms the basis of the modernized IWC and is created based on new technologies.

3. Solution

The general block diagram of the IWC operating algorithm is shown in **Fig. 1**. The block diagram consists of the following blocks:

- Initial parameter determination service;
- M1 – Block for checking the presence of a request from external devices to the controller;
- M2 – Request validity check block;
- M3 – Request service block.

3.1. Initial parameter determination service

Each IWC consists of a control board, a power supply unit, a radio modem intended for radio communication, etc. The IWC is controlled via a microprocessor (MP) placed on the control board. This MP is programmed on a computer using a special **programmer** before being installed in the IWC. During programming, the IWC's operating program is written to the MP. During programming, certain parameters used by the given program are written to the MP's EEPROM (Electrically Erasable Programmable Read-Only Memory) starting from a specific address (**0x08**). These parameters are as follows: the serial number of the IWC where the MP will be placed, the group number of the IWC, the radio communication frequency between the IWC and the radio modem, and the limit value set for the number of points during a dynamometer chart request. Note that coding in the microprocessor is carried out in the hexadecimal system.

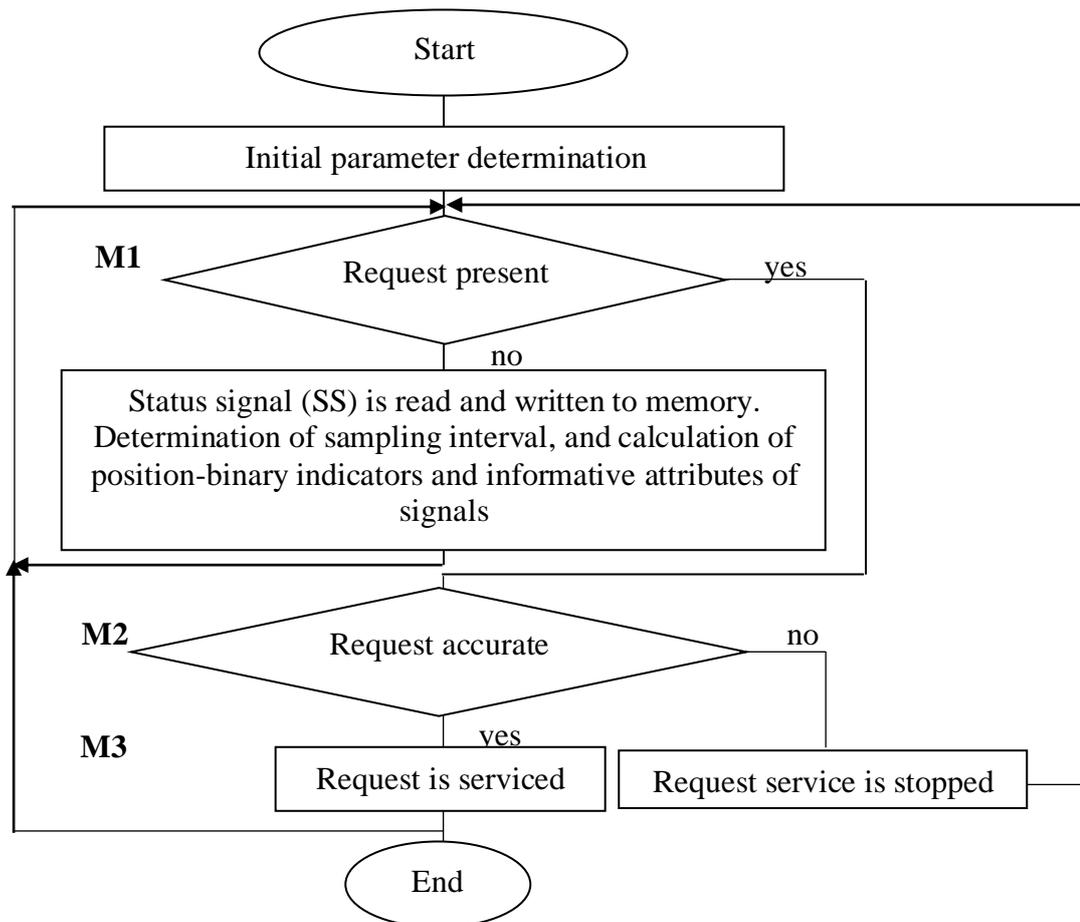


Fig. 1. Operating algorithm of the sucker rod pumping unit well controller

Therefore, the parameters written to the **EEPROM** during programming are recorded in the MP specifically in this numeral system. For instance, suppose that **0A 01 30 04 7D** was written to address **08** of the **EEPROM** during programming. Here, **0A** means the serial number of the IWC is **0x0A=10** (in decimal), and **01** means the group number of the IWC is 1. The subsequent numbers **30** and **04** indicate that the communication frequency of the radio modem is **430 Hz**, while **0x7D=125** (in decimal) shows the limit value set for the number of dynamometer chart points. To save space in the EEPROM, the limit value is taken as one byte, and to find the limit set for the number of dynamometer chart points, it is necessary to multiply the given number by eight. That is, the limit value being **7DH** means the limit set for the number of dynamometer chart points is equal to **1000** points: **since 7DH is equal to 125 in the decimal system, multiplying it by 8 results in 125 x 8 = 1000 points.**

Thus, after the MP is programmed, it is placed into the specific socket intended for it on the IWC board and is ready for use.

When voltage is applied to the IWC, first the program in the MP starts, and the initial parameter determination service is executed within this program. During this process, the MP memory is cleared (reset). The special parameters mentioned above, which are used during the operation of the IWC, are written from the MP's EEPROM memory to the corresponding locations in the RAM (Random Access Memory): the value at EEPROM address **0x08**, indicating the IWC serial number, is assigned to the parameter named **NIWC** in RAM; the value at EEPROM address **0x09**, indicating the IWC group number, is assigned to the parameter named **NGR** in RAM; the value at EEPROM addresses **0x0B:0A**, indicating the RMD communication frequency, is assigned to the parameter named **Frequency** in RAM; and the value at EEPROM address **0x0C**, representing

the limit for the number of dynamometer chart points, is assigned to the parameter named **NLim**.

In accordance with their designations in the IWC, the status of the ports (**PA, PB, PC, PD**) is determined. According to their designation, these ports are divided into input and output ports.

The radio modem is programmed based on the IWC serial number, IWC group number, and Frequency parameters. The information exchange (**UART**) mode with external devices is defined (e.g., **9600 baud, 8 bits, no parity, 1 stop bit**). The **UART** is programmed into the data reception (**RX**) mode. The timer operating mode is defined. The analog-to-digital converter (**ADC**) is programmed into the data reading mode.

3.2. M1 – Block for checking the presence of a request from external devices to the controller

As mentioned above, the IWC operates in a passive cyclic mode based on the algorithm shown in **Fig. 1**. That is, it checks for the presence of a request from the computer. When there is no request, it performs some routine tasks, and when a request occurs, it provides service to it.

Table 1

The protocol for the request to read the content of a specific address or change the content of the address

Request	Response
Target device (IWC) serial number	Source device (computer) serial number
Target device (IWC) group number	Source device group number
Source device (computer) serial number	Target device (IWC) serial number
Source device group number	Target device (IWC) group number
Purpose of request: FA = 3 – Read memory content; FA = 6 – Change memory content	Request designation: FA = 3 – read request; (FA = 6 – write request)
Byte count	Total bytes (4+read/written)
Target address low byte for reading (writing)	Target address low byte for reading (writing)
Target address high byte for reading (writing)	Target address high byte for reading (writing)
Number of bytes to read (write) – low byte	Number of read/written bytes – low byte
Number of bytes to read (write) – high byte	Number of read/written bytes – high byte
Cyclic Redundancy Check (CRC) – low byte	1st byte
Cyclic Redundancy Check (CRC) – high byte	2nd byte

	n-th byte
	Cyclic Redundancy Check (CRC) – low byte
	Cyclic Redundancy Check (CRC) – high byte

The **MODBUS** protocol is used for data exchange (requests) between the computer and the IWC. Requests are generated on the computer and are structured differently depending on their purpose: the request intended for reading or changing the content of a specific address in the IWC (**Table 1**), the request sent to the IWC to receive a dynamometer chart (**Table 2**), and the request intended for issuing a TC signal (**Table 3**).

Before the bytes forming the request are sent from the computer to the IWC, a cyclic redundancy check (**CRC**) is calculated based on them and added to the end of the request. This two-

byte code is used to verify the accuracy (integrity) of the request information during data exchange. The calculation of the CRC is based on finding the remainder obtained when dividing the data forming the request by a specific polynomial (**0xA001**) using a particular algorithm. During data exchange, the request is sent to the IWC along with the CRC. The receiving side (IWC), in turn, recalculates the CRC using the same algorithm and all the bytes it has received. If this recalculated control code is equal to zero, the request is considered to be received completely and correctly; otherwise, it is assumed that the request was either not received in full or has been subject to external interference.

Output port **D.7** is used to check for the presence of a request from external devices (computer) in the IWC. During the initial assignment of the ports, a '0' is written to this port.

A request sent from the computer to the IWC creates an interrupt in the operation of the MP, triggering the UART's reception driver, and the incoming data is received byte-by-byte and written to the IOADR array prepared for them. Once the first byte is received, a '1' is written to port **D.7**.

Table 2

The protocol for the request to receive a dynamometer chart

Request
Target device (IWC) serial number
Target device (IWC) group number
Source device (computer) number
Source device group number
Purpose of request: FA = 1
Number of bytes = 2
Number of dynamometer chart points to be received (low byte)
Number of dynamometer chart points to be received (high byte)
Cyclic Redundancy Check (CRC) – low byte
Cyclic Redundancy Check (CRC) – high byte

Table 3

The protocol for the TC command request

Request
Target device (IWC) serial number
Target device (IWC) group number
Source device (computer) number
Source device group number
Purpose of request : FA = 4 (FA = 5)
Number of bytes = 2
TC command duration (low byte)
TC command duration (high byte)
Cyclic Redundancy Check (CRC) – low byte
Cyclic Redundancy Check (CRC) – high byte

During the periodic operating cycle of the IWC, port **D.7** is checked every time in the main program. A value of '0' at port **D.7** means that there is no request from external devices. In this case, the MP reads the status signal at the B1:B0 ports, i.e., the SS information, and writes the result to the **InfTS** address intended for storing this information. Additionally, during this time, tasks are performed to calculate specific parameters for the early diagnosis of the SRPU (determination of the

sampling interval, calculation of position-binary indicators, and calculation of the informative attributes of the signals). Then, control is returned to the beginning of the cycle.

If during the check it is discovered that the main D.7 port is '1', this means that a request from an external device has either begun to be received by the IWC or has been fully received. In this case, to ensure the request is completely received, an additional waiting period of $\tau=100$ Ms is observed. It is assumed that if a request has started to be received, the given duration is sufficient for it to be received in full. Then the process for checking the request validity of the received request is started.

3.3. M2 – Request validity check block

The algorithm for checking the request validity (integrity) is given in **Fig. 2**. In this process, several parameters are reviewed and verified. The first parameter considered is the verification of the number of received bytes. The goal here is to check whether the reception truly occurred based on a request sent from an external device (computer) or as a result of any external interference (external devices, interference in communication lines, signal distortion, etc.).

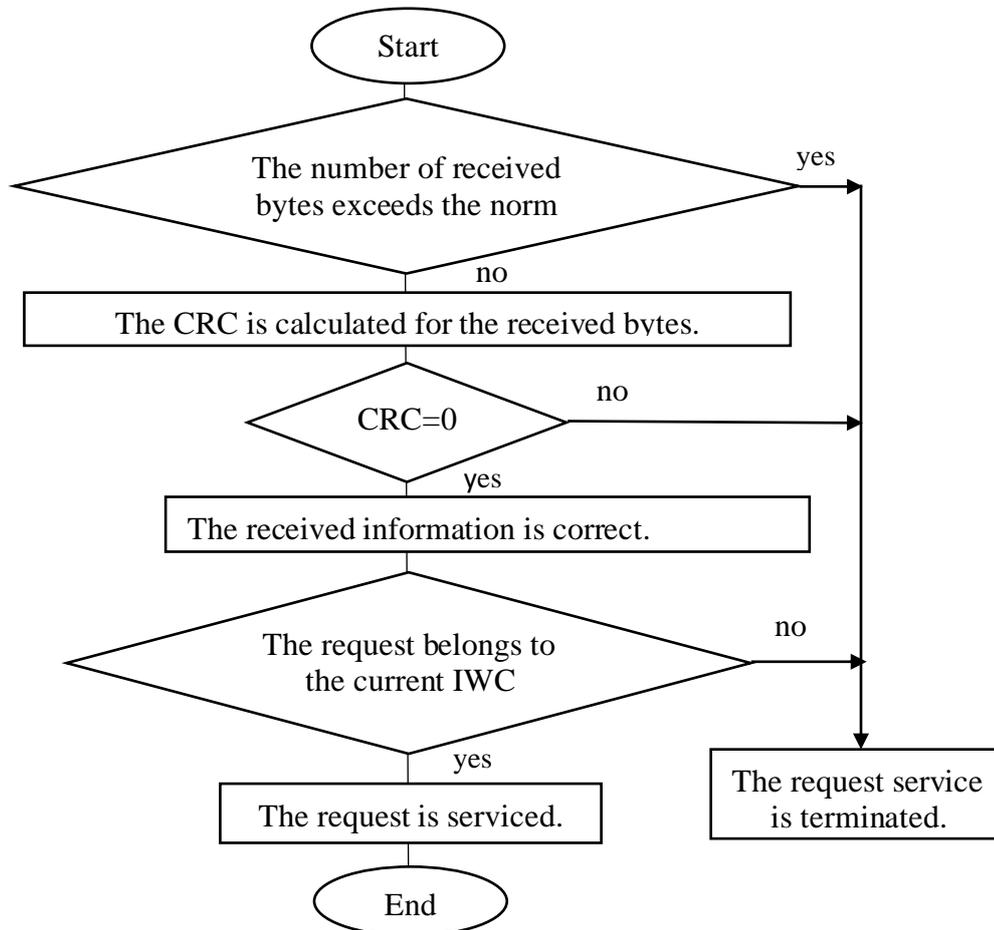


Fig. 2. Request validity check algorithm

Requests sent from the computer to the IWC are known in advance, and none of them exceed a length of $n = 20$ bytes. Therefore, if the number of received bytes does not exceed the given limit n , the process of the request validity check continues; otherwise, it is assumed that the reception occurred as a result of external interference. In this case, the request service is terminated.

The second parameter considered in the process of checking the request validity of the

received data is the verification of the cyclic redundancy check. After the request is received by the IWC, the received bytes are located in the IOADR array. The CRC is recalculated for all elements of this array. If this code is equal to zero, the request is considered correct. Otherwise, the request is considered incorrect, and the request service is terminated.

The last parameter checked is whether the correct received request belongs to the current IWC or not. The “**target device serial number**” located at the IOADR address of the request (counting starts from 0) and the “**target device group number**” located at the IOADR+1 address are retrieved. If the target device serial number is equal to the current IWC serial number (NIWC) and the target device group number is equal to the current IWC group number (NGR), then the received request belongs to the current IWC, and the process moves to the stage of providing request service to the given request. Otherwise, i.e., if at least one of the above two comparisons does not result in equality, it is assumed that the given request does not belong to the current IWC, and the request service is terminated.

3.4. M3 – Request service block

The algorithm for the request service block is given in **Fig. 3**. The main parameter considered here is the purpose of the request, i.e., determining the purpose of the request was sent. The byte (**FA**) indicating the purpose of the given request is read from the IOADR array where the request data is stored. This byte is located at the IOADR+4 address: **FA = (IOADR+4)**. Then, the tasks to be performed are determined according to the purpose, i.e., based on the value of FA.

If **FA=1**, it means the request was sent to get a dynamometer chart. In this case, the number of points required to get the dynamometer chart, **D_Num**, must be determined from the IOADR array. This count is formed from the bytes located at address **IOADR+6** (low byte) and address **IOADR+7** (high byte) of the given array: **D_Num = (IOADR+7) *256+(IOADR+6)**. The given number of points is compared with the limit value (**NLim**) set for the number of points as shown above. If the number of points exceeds the given limit, the number of points is set equal to the limit value (e.g., if the number of points in the sent request is greater than 1000, 1000 points are taken as the number of points); otherwise, that is, if the number of points does not exceed the limit value, it is left unchanged, and the process moves to acquiring the dynamometer chart.

Dynamometer chart data – the output parameters of the stroke and force transmitters – are read from the ADC. The volume of the read parameters depends on the resolution of the ADC (e.g., 10 bits, 16 bits, or 24 bits). Since the resolution of the ADC used is known in advance, an appropriate amount of space is allocated for the read parameters (2 bytes or 3 bytes for the ADC resolutions shown above). The read parameters are written byte-by-byte to the **UART** output register (**UDR**), meaning they are sent to the computer via radio communication. Then, in accordance with the requirements of the radio communication, a certain pause is provided (for example, 70 milliseconds) to ensure the sent bytes reach their destination. After that, the ADC is accessed to read the dynamometer chart data for the next point. Then a pause is given again, and so on. The process continues until the dynamometer chart information for the specified number of points (**D_Num**) is read. Once the specified number of dynamometer chart data points have been read and sent to the computer, the dynamometer chart acquisition process ends. Then, the MP is set to a ready state to receive the next request: the UART is set to reception mode, the IOADR array and the counter indicating the number of received points are reset, and port **D.7** is set to the ‘0’ state. These tasks are performed at the end of each request service process.

If **FA=4** or **FA=5**, it means the request was sent for a **TC** command (or multiple TC commands). If **FA=4**, the TC1 command must be executed; if **FA=5**, the TC2 command must be executed. The TC command duration is retrieved from address **IOADR+6** (low byte) and address **IOADR+7** (high byte) of the IOADR array and written to the **TC_Time** counter: **TC_Time =**

$(\text{IOADR}+7) * 256 + (\text{IOADR}+6)$. Here, the time is expressed in seconds. Then, the corresponding TC command is issued: **PortC,4** (or **PortC,5**) = '1'.

After the TC command is issued, the duration specified in the **TC_Time** counter is observed. As soon as the given time expires, the TC command is deactivated: **PortC,4** (**PortC,5**) = '0'. This concludes the process of executing the TC command. Then, the MP is set to a ready state to receive the next request.

If **FA=3**, it means the request was sent to read the contents of a given number of addresses starting from a specific address in the MP's RAM and send them to the computer. The starting address from which the data will be read is located at address **IOADR+6** (low byte) and address **IOADR+7** (high byte) of the **IOADR** array: **Address** = $(\text{IOADR}+7) * 256 + (\text{IOADR}+6)$. The number of bytes to be read from the specified address is located at address **IOADR+8** (low byte) and address **IOADR+9** (high byte): **Read_Count** = $(\text{IOADR}+9) * 256 + (\text{IOADR}+8)$. Then, the destination address where the data read from the given address will be written is determined: **Write_Address** = $(\text{IOADR}+10)$.

Thus, a specific number of bytes (**Read_Count**) are read from the given (**Address**) address of the RAM and written to the **Write_Address** address: (**Write_Address+i**): = (**Address+i**).

After the process of reading the **content** is completed, preparations are made for sending the data to the computer.

3.5. Sending the data to the computer

First, a transition is made from the request protocol to the response protocol. For this purpose, the positions of the bytes indicating the **target** and **source** device numbers at the beginning of the MODBUS protocol are swapped. Now, the target device is the computer, and the source device is the IWC. The first four bytes in the protocol header are arranged as follows (in accordance with the requirements of the MODBUS protocol):

Target device (Computer) serial number
Target device (Computer) group number
Source device (IWC) serial number
Target device (IWC) group number.

Then, the byte count located at address $(\text{IOADR} + 5)$ is increased by the number of read (written) bytes and written back to that location:

Byte count: = **Byte count** + **Read_Count** (**Write_Count**).

Finally, the CRC is calculated for all bytes located in the **IOADR** array and added to the end of the protocol.

Thus, the **IOADR** array is ready to be sent to the computer. The number of bytes to be sent to the computer (the number of elements in the **IOADR** array) is determined. Then, the elements of the **IOADR** array are written one by one to the **UART** output register (**UDR**), meaning they are sent to the computer via radio communication. After all bytes have been sent to the computer, a certain pause is provided (e.g., 70 Ms), and the request service process is completed. Then, the MP is set to a ready state to receive the next request.

If **FA=6**, it means the request was sent to change the content of a specific number of addresses in the RAM. The start of the address whose content will be changed is located at address **IOADR+6** (low byte) and address **IOADR+7** (high byte) of the **IOADR** array: **Address** = $(\text{IOADR}+7) * 256 + (\text{IOADR}+6)$. The number of addresses whose content is to be changed is located at address **IOADR+8** (low byte) and address **IOADR+9** (high byte):

Write_Count = $(\text{IOADR}+9) * 256 + (\text{IOADR}+8)$.

The new content for the above addresses, that is, the new data to be written to the given **Address**, is located at the following address:

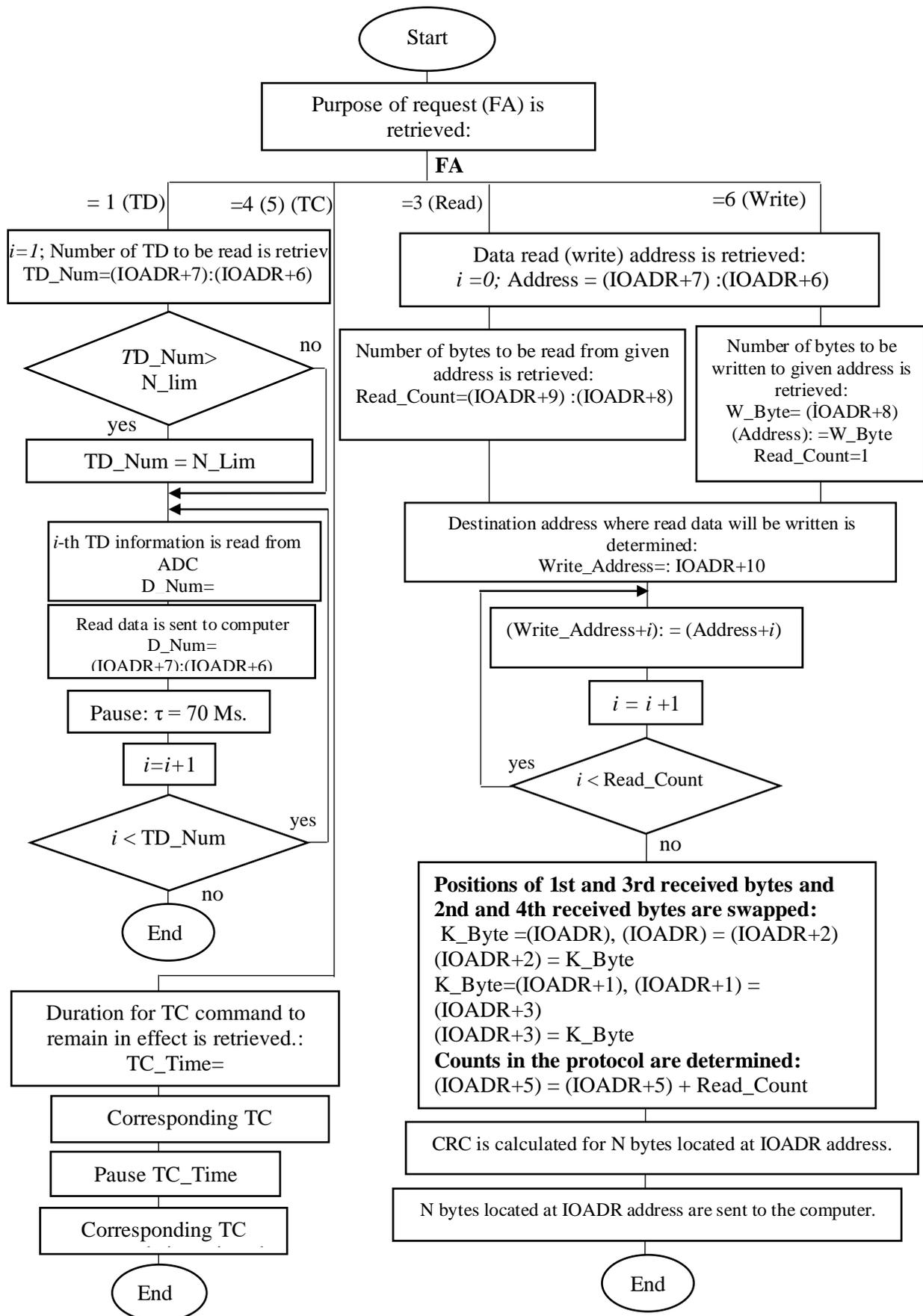


Fig. 3. Request service module algorithm

Write_Address = (IOADR+10).

Thus, **Write_Count** amount of data is taken from the given **Write_Address** of the RAM and written to the **Address**:

(Address+ i): = (Write_Address+i).

After the process of changing the address content is completed, preparations are made for sending the data to the computer. This process is identical to the process of reading the address content. Therefore, a transition is made to the **Sending the data to the computer** section.

4. Conclusion

An operating algorithm for controllers for the operational monitoring and control of sucker rod pumping unit operating modes is developed. As a result, it becomes possible to monitor the operating modes of oil wells operated by beam pumping units, provide early diagnosis of malfunctions that may occur in the technical condition of mechanical equipment, and implement the sub-optimal operating modes of the well in "on-line" mode. This, in turn, increases the functionality and technical capabilities of the overall oil field control system.

References

- [1] И.А. Ниссенбаум, Р.А. Кудряшов, Ю.Б. Новоселов, В.Т. Фрайштетер, Современное состояние проблемы энергосбережения на нефтяных промыслах Тюменской области, Энергетика Тюменского Региона. No.3 (2000) pp.2-9. [In Russian: I.A. Nissenbaum, R.A. Kudryashov, Y.B. Novoselov, V.T. Fraishteter, Current state of the energy saving problem in the oil fields of the Tyumen region, Energy of the Tyumen Region].
- [2] Ас.Г. Рзаев, М.Г. Резван, М.И. Хакимьянов, И.Н. Шафиков, Современное состояние автоматизации установок механизированной добычи нефти на территории СНГ, Известия НАНА, серия физико-технических и математических наук. 33 No.6 (2013) pp.176-186. [In Russian: As.H. Rzayev, M.G. Rezvan, M.I. Khakimyanov, I.N. Shafikov, Current state of automation of mechanized oil production units in the CIS territory, News of ANAS, Series of physical-technical and mathematical sciences].
- [3] Т.А. Алиев, О.Г. Нусратов, Г.А. Гулиев, Ас.Г. Рзаев, Ф.Г. Пашаев, М.Г. Резван, Позиционно-бинарная технология обработки сигналов усилия для идентификации технического состояния установок штанговых глубинных насосов, Журнал измерительная техника, приложение «Метрология». No.3 (2018) pp.14-24. [In Russian: T.A. Aliev, O.G. Nusratov, G.A. Guliyev, As.H. Rzayev, F.H. Pashayev, M.G. Rezvan, Position-binary technology for processing force signals to identify the technical condition of sucker rod pump units, Journal of Measurement Techniques, Metrology supplement].
- [4] Ас.Г. Рзаев, Я.Г. Алиев, М.Г. Резван, Повышение частоты дискретизации выходных сигналов датчиков динамометрирования ДУИ и ДУП, Электротехнические Комплексы и Системы, Международная научно-практическая конференция, изд. УГАТУ. (2020) pp.8-13. [In Russian: As.H. Rzayev, Y.G. Aliyev, M.G. Rezvan, Increasing the sampling frequency of output signals from DUI and DUP dynamometer sensors, Electrotechnical Complexes and Systems, International Scientific and Practical Conference, USATU Publishing].
- [5] Г.А. Гулуев, Ас.Г. Рзаев, М.Г. Резван, Технология и алгоритм вычисления индикаторов для ранней диагностики неисправностей штанговых глубинных насосов, Электротехнические Комплексы и Системы, Международная научно-практическая конференция, изд. УГАТУ. (2020) pp.42-48. [In Russian: G.A. Guluyev, As.H. Rzayev, M.G. Rezvan, Technology and algorithm for calculating indicators for early diagnostics of sucker rod pump malfunctions, Electrotechnical Complexes and Systems, International Scientific and Practical Conference, USATU Publishing].
- [6] Т.А. Алиев, Г.А. Гулуев, Ас.Г. Рзаев, М.Г. Резван, Алгоритм позиционно-бинарной технологии контроля начала и динамики развития аварий штанговых глубинно-насосных установок, Известия НАНА, серия физико-технических и математических наук, Баку. 42 No.2 (2022) pp.54-64. [In Russian: T.A. Aliev, G.A. Guluyev, As.H. Rzayev, M.G. Rezvan, Algorithm for position-binary technology to monitor the onset and dynamics of accident development in sucker rod pumping units, News of ANAS, series of physical-technical and mathematical sciences, Baku].