Communication Protocol for Decoders of the 4th Generation T42, T45, T46 and indicators T42

1. General Terms

The subset of the **Modbus Application Protocol** ***(Modbus RTU)*** for networks based on RS 485 physical layer protocol is implemented in T46 decoders and T42 (RS-485) indicator. Unlike the standard Modbus protocol this subset does not support the "**Discrete Inputs"** data type and does not support the following standard commands:

- **1 (0x01)** — read values from several coils *(Read Coil Status)*.
- **2 (0x02)** — read values from several discrete inputs *(Read Discrete Inputs)*.
- **15 (0x0F)** — record values to several coils *(Force Multiple Coils)*
- **22 (0x16)** — record to one holding register using "AND" mask and "OR" mask *(Mask Write Register).*
- Commands with codes **7, 8, 11, 12, 20, 21**.

The similar protocol is implemented in T42, T45 decoders and indicators T42(USB), T42(RS-232), T42 (Ethernet). But they do not have the device address in the query and response titles. T45 decoders and T42(USB), T42(Ethernet) indicators do not have the error checking field because the data integrity is provided with USB and TCP/IP, respectively.

1.1. Decoder Registers

1) **Flag registers (*Coils*)** are one-bit registers with 16-bit address between 0 and 3. They are used for reading and recording.

2) **Holding registers (*Holding Registers*)** are 16-bit registers with addresses between 0 and 4. They are used for reading and recording.

3) **Input registers (*Input Registers*)** are 16-bit registers with addresses between 0 and 17. They are used for reading only.

See the detailed description of the registers in the table below.

Coils		
Name	**Address**	**Size**
Start/Stop measurements (StartStop) determines the start and stop of measuring.	0	1 bit
Streaming (StreamingTransfer) determines the data transfer (communication) mode with a decoder StreamingTransfer = 0 – query-response mode (dialog mode). Available for all decoders; StreamingTransfer = 1 – streaming mode. The decoder automatically transfers data to the host computer as the buffers are full. The purpose is to transfer the entire flow of measurements to the host computer. Available for T42, T45 decoders and T42(USB), T42(RS-232), T42(Ethernet) indicators.	1	1 bit
External speed transducer (ExternalRFT) additional equipment for measuring of the rotating speed at low revs.	2	1 bit
Using floating-point numbers (UsingFloat) for torque (force) and rotating speed values.	3	1 bit
Holding registers		
Name	**Address**	**Size**
Configuration Flags (ConfigWord) is a word with flags registers. The bit number in the word matches the address of the flag.	0	16 bits
Averaging factor /coefficient (AveragingFactor) is used for calculating the value of torque (force). Specifies the number of consecutive measurements that are added up. The average value is obtained by dividing the sum by AveragingFactor.	1	16 bits
Speed measurement time interval ms (SpeedMeasurementPeriod) - reserved	2	16 bits
Current decoder time - low part (TimeLow)	3	16 bits
Current decoder time - high part (TimeHigh)	4	16 bits

Input registers		
Name	**Address**	**Size**
Current average torque (force) – word 1: 1. If UsingFloat=0 – the significant part of a fixed-point number (MomentInt); 2. If UsingFloat=1 – low part of 32-bit floating-point number (MomentLow).	0	16 bits
Current average torque (force) – word 2: 1. If UsingFloat=0 – position of a fixed point (power of 10) for a fixed-point (MomentExp). Formula for calculating the value of torque (force): ***Moment = MomentInt*10^MomentExp.;*** 2. If UsingFloat=1 – the high part of the 32-bit floating-point number (MomentHigh).	1	16 bits
Current rotating speed – word 1: 1. If UsingFloat=0 – the significant part of a fixed-point number (RotationFrequencyInt); 2. If UsingFloat=1 – the lower part of the 32-bit floating-point number (RotationFrequencyLow)	2	16 bits
Current rotating speed – word 2: 1. If UsingFloat=0 – position of a fixed point (power of 10) for a fixed-point (RotationFrequencyExp). Formula for calculating the value of rotating speed: ***RotationFrequency = RotationFrequencyInt*10^RotationFrequencyExp;*** 2. If UsingFloat=1 – the high part of the 32-bit floating-point number (RotationFrequencyHigh).	3	16 bits
Current temperature – the significant part of a fixed-point number (TemperatureInt). Position of a fixed point (power of 10) = -1. Formula for calculating the value of temperature: ***Temperature = TemperatureInt/10.***	4	16 bits
Decoder state (Station). The status word uses two bits: Bit 0 – Transducer connected; Bit 2 – Service information received	5	16 bits
Message counter (MessagesCount)	6	16 bits
Messages – up to 10 pcs (Messages[10]) Messages are a way to inform the host computer about the events occurring at random times. Each event has its own number. The message codes are listed in the Appendix. The decoder action depends on the value of the "Streaming" flag: 1. If StreamingTransfer=0, messages are placed in this buffer and the message counter is incremented. When the messages are read by the main computer, the buffer is cleared and the counter is reset. When the buffer overflows, it starts to fill up again. 2. If StreamingTransfer=1, messages are automatically transferred to the host computer. The buffer is cleared.	7 - 16	16 bits
Firmware Version Number (VersionNumber)	17	16 bits

1.2. Calculation of measured values

If the floating-point numbers are used (UsingFloat=1), the measured value is a 32-bit word that is located in the input registers under the address:

- 0 – for torque (force)
- 2 – for rotating speed

If the fixed-point numbers are used (UsingFloat=0) the measured value is calculated with the formula

- **Torque/ force** (unit of measurement) = MomentInt * 10^MomentExp;
- **Rotation speed (rpm) =** RotationFrequencyInt * 10^RotationFrequencyExp;

Formula for calculating the value of temperature: **Temperature** (°C) = TemperatureInt / 10;

In both cases the unit of measurement for torque (force) is the same and depends on the nominal torque (force) only. It is recognized by the decoder automatically.
See the possible units of measurement in the table below:

Nominal torque (force)	Unit of measurement of the received data
0.1 Nm (N) ÷ 8 Nm (N)	mNm (mN)
10 Nm (N) ÷ 8 kNm (N)	Nm (N)
10 kNm (N) и более	kNm (kN)

1.3. Transducer Zero Adjustment

Torque and force transducers as received or during the mounting on the test bench, may have a “zero shift”.

In this regard, the developer of the data acquisition system should provide option of zero adjustment and saving of the corrections.

The measurement data received from a transducer should be corrected by subtracting the correction value (i.e. zero shift value). The correction value can be also used to take the tare weight from the gross weight.

2. Interactive Mode of Operation

2.1. The General Query and Response Format

All queries and responses have the following format:

Title	Data	Checksum

- Title;
- Data;
- Checksum.

2.2. Title and Checksum Format for T46 Decoders and T42(RS-485) indicators.

For T46 decoders the query and response titles have the following format:

Device address	Command code

- The device address (1 byte) is the address of the device which the query is addressed to (it is useful for the T46 decoder only). The device address can be changed from 1 to 247. The addresses in the range between 248 and 255 are reserved.
- The command code (1 byte) tells the slave device which data or which execution the master device requires. The high unit (0 x 80) in the command code is used in the slave device response to point that the operation is failed;

The checksum (2 bytes)

Checksum

The checksum verifies the absence of errors in the query and response. This is the cyclic checksum CRC-16 Modbus.

2.3. Title and Checksum Format for T42, T45 Decoders and T42(USB), T42(RS-232), T42(Ehternet) Indicators

For T45 decoders and T42(USB), T42(RS-232), T42(Ehternet) indicators the query and response titles have the following format.

Query title:

Command code

- Command code (1 byte);

There is no checksum for T45 decoders and T42(USB), T42(Ehternet) indicators.
There is a checksum for T42 decoders and T42(RS-232) indicators.

2.4. Decoder Command Codes

All the decoders have identical command set:

- 3 – **READ_HOLDING_REGISTERS** – read values from several holding registers
- 4 – **READ_INPUT_REGISTERS** – read values from several input registers
- 5 – **FORCE_SINGLE_COIL** – record single flag value
- 6 – **PRESET_SINGLE_REGISTER** – record value to one holding register
- 16 – **PRESET_MULTIPLE_REGISTERS** – record values to several holding registers
- 17 – **REPORT_SLAVE_ID** – read service information about a device

2.5 The Field Format of the Queries

- **READ_HOLDING_REGISTERS (3)** ***query***
  - Address of the first item in the holding register table (2 bytes);
  - Number of readable registers (2 bytes).
- **READ_INPUT_REGISTERS (4)** ***query***
  - Address of the first item in the holding register table (2 bytes);
  - Number of readable registers (2 bytes).
- **FORCE_SINGLE_COIL (5)** ***query***
  - Register address (2 bytes);
  - Set value (2 bytes). The value 0xFF00 means the on-mode for flags, value 0x0000 means the off-mode. Other values are invalid.
- **PRESET_SINGLE_REGISTER (6)** ***query***
  - Register address (2 bytes);
  - Set value (2 bytes).
- **PRESET_MULTIPLE_REGISTERS (16)** ***query***
  - Register address (2 bytes);
  - Number of recordable registers (2 bytes);
  - Number of forwarded(transferred) bytes - N (1 byte);
  - Set values (N bytes).
- **REPORT_SLAVE_ID (17)** ***query***

  Data field is empty.

2.6. The Field Format of the Responses

If an error occurs during the command execution the decoder returns the completion code (1 byte) to the data field. The completion code can take the following values:

- **RET_CODE_COMMAND_ERR (1)** – received function code cannot be processed.
- **RET_CODE_ADDRESS_ERR (2)** – data address specified in the query is not available.
- **RET_CODE_DATA_ERR (3)** – value in query data field is invalid.
- **RET_CODE_UNREPAIRABLE_ERR (4)** – unrepairable error occurred while the slave device was attempting to execute the requested action.
- **RET_CODE_BUSY_ERR (6)** – the slave device is busy with the command processing. The master device should repeat the message later, when the slave device is free.
- **RET_CODE_CHECKSUM_ERR (8)** – the slave device has detected a parity error.

The error occurred during the command execution is indicated with 1 in the high-order byte of the command code in the response title.

The response data formats if the operation is successful are given below:

- ***Response to*** **READ_HOLDING_REGISTERS (3)**
  - Number of forwarded bytes - N (1 byte);
  - Requested registers (N byte);

- ***Response to*** **READ_INPUT_REGISTERS (4)**

  - Number of forwarded bytes - N (1 byte);
  - Requested registers (N byte);

- ***Response to*** **FORCE_SINGLE_COIL (5)**

  - Register address (2 bytes);
  - Set values (2 bytes). The value 0xFF00 means the on-mode for flags, value 0x0000 means the off-mode. Other values are invalid.

- ***Response to*** **PRESET_SINGLE_REGISTER (6)**

  - Register address (2 bytes)
  - Set values (2 bytes).

- ***Response to*** **PRESET_MULTIPLE_REGISTERS (16)**

  - Register address (2 bytes);
  - Number of recordable registers (2 bytes);

- ***Response to*** **REPORT_SLAVE_ID (17)**

  - Transducer ID (3 bytes);
  - Temperature (Integer 1 byte);
  - Sensitivity correction (Integer 1 byte);
  - Teeth number (Integer 2 bytes);
  - Max rotating speed divided by 100 (Integer 1 byte);
  - Check date (3 bytes);
  - Additional text information (49 bytes);

2.7. Transfer Byte Order in Words

According to the Modbus protocol the T46 decoder and T42(485) indicator receive queries and forward responses where the first byte is the **high byte.**

Other decoders (T42, T45) and indicators (T42(USB), T42(RS-232), T42 (Ehternet)) receive queries and forward responses where the first byte is the **low byte.**

2.8. Queries and Responses of the T46 Decoder and T42(RS485) Indicator. Samples

- **FORCE_SINGLE_COIL (5)** ***command***
  Set the "Start/Stop" configuration bit in 1, i.e. start measurements.
  Device address = 1.
  Register address in the holding registers partition =0x0000
  Word for record = FF00

```
Query: 01 05 0000 FF00 8C3A
Response:  01 05 FF00 5029
```

- **PRESET_SINGLE_REGISTER (6)** ***command***
  Set the average factor equal 100.
  Device address = 1.
  Register address in the holding registers partition =0x0001
  New value = 0x0064

```
Query: 01 06 0001 0064 D9E1
Response: 01 06 0001 0064 D9E1
```

- **PRESET_MULTIPLE_REGISTERS (16)** ***command***
  Set the time on the decoder clock equal 0.
  Device address = 1.
  Register address in the holding registers partition=0x0003
  Registers number = 2
  New values are: low part = 0x0000, high part = 0x0000.

```
Query: 01 10 0003 0002 04 0000 0000 B3BA
Response:  01 10 0003 0002 B1C8
```

- **READ_HOLDING_REGISTERS (3)** ***command***
  Read the current time on the decoder clock.
  Device address = 1.
  Register address in the holding registers partition=0x0003
  Registers number – 2

```
Query: 01 03 0003 0002 340B
Response:  01 03 04 B0C1 002E 0D13
```

- **READ_INPUT_REGISTERS (4)** ***command***
  Read 5 input registers, starting from the address 0.
  Device address = 1.
  Register address in the holding registers partition=0x0000
  Registers number – 2

```
Query: 01 04 0000 0005 3009
Response:  01 04 0A 0FA0 0000 0E4F FFFE 012C 1C03
```

- **REPORT_SLAVE_ID (17)** ***command***
  Read the service information of a transducer.
  Device address = 1.

```
Query: 01 11 C02C
Response:  01 11 043500A07F01003201050CD3E0F0E8F0EEE2E0EB20C1F3EBE0E2EAEE20C3
              E8F1F2E5F0E57E8F120302C312520D2EEF7EEDEEF1F2FC20302C31352500 245E
```

3. Streaming Mode of Operation

The T42, T45 decoders and T42(USB), T42(RS-232), T42 (Ethernet) support the streaming mode of operation. This mode is used to forward the whole measurements flow to the mainframe.

To activate the streaming mode set flags “Start/Stop” and “Streaming” in 1. The decoder will automatically buffer from several torque (force) measurements and transfer them to the mainframe. The size of the buffer depends on the decoder type and on the averaging factor.

If you use rotating transducers, according to the computational algorithm, current rotating speed will be transferred to the mainframe several times per second.

Temperature is transferred as its change is detected.

For decoders with firmware number up to 19 (inclusive) in streaming mode torque (force) is transferred as fixed-point numbers only. The fixed-point position is the same for all measurements and can be read from the «MomentExp» input register. The torque (force) calculating formula is the same as in the interactive mode. Temperature and rotation speed are transferred as floating-point numbers and do not require any additional calculations.

For decoders with firmware number up to 20 and more in streaming mode Torque (force) can be transferred both as fixed-point numbers and floating-point numbers. This depends on the value of the UsingFloat flag. Buffer size for numbers with floating point is the same, but the number of measurements is half that for the same values of averaging factor (AveragingFactor).

T45 decoder and T42(Ethernet) indicator let you use the streaming and the interactive modes in parallel, it uses duplex line. T42 decoder and T42(RS-232) indicator have half-duplex line. To activate the interactive mode you should set off the streaming or stop all measurements. To send the command to the decoder (indicator) in this case, you need to suspend streaming. To pause the oncoming transmission, set 1 on the RTS line. To renew transmission, set 0 on the RTS line.

3.1 Buffer Format for Streaming to the Mainframe

General buffer format:

Title	Data	Checksum

The checksum is in T42 decoders and T42(RS-232) indicators only.

Title format:

Data type	Data length (2 bytes)

Where: Data type:

- **100 – torque (force)**. Data structure:

```
//--------------------- Torque (force) buffer format
struct _MeasuresBuffer {
short int  TimeLow;          // Time from the start of measurements – low part
short int  TimeHigh;         // Time from the start of measurements – high part
unsigned char BufferCount;   // Buffer number
unsigned char DataCount;     // Number of measurements in buffer
short int  Moments[DataCount]; // Moment in fixed-point format
(or float Moments [DataCount]; // Moment in floating point format)
};
NOTE: Buffer structure description is conditional.
```

- **101 – rotating speed**. Data structure:

```
//--------------------- Rotating speed buffer format
struct _RotationFrequency {
  unsigned short int  TimeLow;
  unsigned short int  TimeHigh;
  float RotationFrequency;
};
```

- **102 – temperature**. Data structure:

```
//---------------------- Temperature buffer structure
struct _TemperaturaBuffer {
  short int  TimeLow;
  short int  TimeHigh;
  float Temperatura;
};
```

- **103 – messages**. Data structure:

```
//---------------------- Messages buffer format
struct _MessageBuffer {
  short int  TimeLow;
  short int  TimeHigh;
  short int  MessageCount;
  short int  Messages[MessageCount];
};
```

4. Transducers Synchronization

If your measurement system has several connected transducers with 4th generation decoders, you can synchronize data received from different transducers. Use decoders' clock to do it.

Each data block is provided with a time stamp when this block was formed. The time interval between two measurements is 0.0002 sec, if the averaging coefficient is "1". If you set the decoder clock to the time of the main computer (microcontroller), you can match data with the same time stamp. The computer (microcontroller) receives data at different time, but these data are displayed on a graph with the same position on the time axis.

Technical implementation:

- At the program start (before starting of measurements and setting the streaming mode), choose any time of microcontroller T0 in seconds and save it.
- Take the time of microcontroller T1 and compute the time difference (DT) in seconds: DT1 = T1 – T0. Convert this relative time into units of decoder clock according to the following formula:
  DT1_d = DT1*62500. Where 62500 denotes the decoder clock frequency.
- Write this 32-bit number into two 16-bit words TimeLow and TimeHigh of the first decoder.
- Repeat this operation with all transducers. Calculate the relative time DT1_d.
- To avoid the decoder clock offset repeat this operation at any moment, approximately once per four seconds.

NOTE! The time of microcontroller can be in other units. In this case, the converting formula will be changed: DT1_d = (DT1*62500) / (microcontroller clock frequency).

5. Appendix 1. Message Codes

```
#define MESSAGE_POLUCHEN_ID       2  // message "Service information received!"
#define MESSAGE_OSHIBKA_ID        3  // message "Service information does not match
the old"
#define MESSAGE_DATCHIK_OFF       4  // Transducer disconnected
#define MESSAGE_DATCHIK_ON        5  // Transducer connected
#define MESSAGE_BUFFER_LOSS       15 // Loss of data buffer
#define MESSAGE_MESSAGEQ_OVERFL   19 // Message queue overflow
```