

Протокол обмена информацией для декодеров T42, T45, T46, T47

1. Общие положения

В декодерах T46 реализовано подмножество протокола **Modbus Application Protocol (Modbus RTU)** для сетей основанных на протоколе физического уровня RS-485. В отличие от стандартного протокола Modbus, это подмножество не поддерживает тип данных «дискретные входы (*Discrete Inputs*)» и не поддерживает следующие стандартные команды:

- **1 (0x01)** – чтение значений из нескольких регистров флагов (*Read Coil Status*).
- **2 (0x02)** – чтение значений из нескольких дискретных входов (*Read Discrete Inputs*).
- **15 (0x0F)** – запись значений в несколько регистров флагов (*Force Multiple Coils*)
- **22 (0x16)** – запись в один регистр хранения с использованием маски "И" и маски "ИЛИ" (*Mask Write Register*).
- А также команды с кодами **7, 8, 11, 12, 20, 21**.

В декодерах T42, T45, и T47 реализован похожий протокол. Но, поскольку эти декодеры не предназначены для работы в сети, то у них в заголовке запросов и ответов отсутствует адрес устройства. Кроме этого у декодеров T45 и T47 отсутствует поле контроля ошибок, так как целостность данных обеспечивают USB и TCP/IP соответственно.

1.1. Регистры декодеров

1) **Регистры флагов (Coils)** – однобитовые регистры с 16-разрядным адресом от 0 до 3. Предназначены для чтения и записи.

2) **Регистры хранения (Holding Registers)** – 16-разрядные регистры с адресами от 0 до 4. Предназначены для чтения и записи.

3) **Регистры ввода (Input Registers)** – 16-разрядные регистры с адресами от 0 до 17. Предназначены только для чтения.

В таблице приведено подробное описание регистров декодеров.

| Регистры флагов | | |
|---|-------|-------------|
| Название | Адрес | Разрядность |
| Старт/Стоп измерений (StartStop) . Определяет запуск и останов измерений. | 0 | 1 бит |
| Потоковая передача (StreamingTransfer) . Определяет режим обмена данными с декодером: StreamingTransfer=0 – режим запрос-ответ (диалоговый режим). Доступен для всех декодеров; StreamingTransfer=1 – режим потоковой передачи. Декодер автоматически передает данные в главный компьютер по мере заполнения буферов. Назначение – передать весь поток измерений в главный компьютер. Доступен для декодеров T42, T45 и T47. | 1 | 1 бит |
| Внешний датчик скорости (ExternalRFT) – дополнительное оборудование для измерения частоты вращения при малых оборотах. | 2 | 1 бит |
| Использование чисел с плавающей точкой (UsingFloat) для значений крутящего момента (силы) и частоты вращения. | 3 | 1 бит |
| Регистры хранения | | |
| Название | Адрес | Разрядность |
| Флаги конфигурации (ConfigWord) – слово с регистрами флагов. Номер бита в слове соответствует адресу регистра флага. | 0 | 16 битов |
| Коэффициент усреднения (AveragingFactor) для вычисления значения крутящего момента (силы). Задаёт количество последовательных измерений, которые складываются. Усредненное значение получается путем деления суммы на AveragingFactor. | 1 | 16 битов |
| Период измерения скорости мс (SpeedMeasurementPeriod) – зарезервировано | 2 | 16 битов |
| Текущее время декодера – младшая часть (TimeLow) | 3 | 16 битов |
| Текущее время декодера – старшая часть (TimeHigh) | 4 | 16 битов |

| Регистры ввода | | |
|--|--------|-------------|
| Название | Адрес | Разрядность |
| Текущий усредненный крутящий момент (сила): 1. Если UsingFloat=0 – значащая часть числа с фиксированной точкой (MomentInt) 2. Если UsingFloat=1 – младшая часть 32-разрядного числа с плавающей точкой (MomentLow) | 0 | 16 битов |
| Текущий усредненный крутящий момент (сила): 1. Если UsingFloat=0 – позиция фиксированной точки (степень 10) для числа с фиксированной точкой (MomentExp). Формула для вычисления значения крутящего момента (силы): $Moment = MomentInt * 10^{MomentExp}$. 2. Если UsingFloat=1 – старшая часть 32-разрядного числа с плавающей точкой (MomentHigh) | 1 | 16 битов |
| Текущая частота вращения: 1. Если UsingFloat=0 – значащая часть числа с фиксированной точкой (RotationFrequencyInt) 2. Если UsingFloat=1 – младшая часть 32-разрядного числа с плавающей точкой (RotationFrequencyLow) | 2 | 16 битов |
| Текущая частота вращения: 1. Если UsingFloat=0 – позиция фиксированной точки (степень 10) для числа с фиксированной точкой (RotationFrequencyExp). Формула для вычисления значения частоты вращения: $RotationFrequency = RotationFrequencyInt * 10^{RotationFrequencyExp}$. 2. Если UsingFloat=1 – старшая часть 32-разрядного числа с плавающей точкой (RotationFrequencyHigh) | 3 | 16 битов |
| Текущая температура – значащая часть числа с фиксированной точкой (TemperatureInt). Позиция фиксированной точки (степень 10) = -1. Формула для вычисления значения температуры: $Temperature = TemperatureInt / 10$. | 4 | 16 битов |
| Состояние декодера (Station). В слове состояния используются два бита: Бит 0 – Датчик подключен; Бит 2 – Получена служебная информация | 5 | 16 битов |
| Счетчик сообщений (MessagesCount) | 6 | 16 битов |
| Сообщения – до 10 шт. (Messages[10]). Сообщения – это способ информировать главный компьютер о событиях, происходящих в случайные моменты времени. Каждое событие имеет свой номер. Коды сообщений приведены в приложении. Поведение декодера зависит от значения флага «Потоковая передача»: 1. Если StreamingTransfer =0, сообщения помещаются в данный буфер и счетчик сообщений увеличивается. При считывании сообщений главным компьютером буфер очищается и счетчик обнуляется. При переполнении буфер начинает заполняться заново. 2. Если StreamingTransfer =1, сообщения автоматически передаются в главный компьютер. Буфер при этом очищается. | 7 - 16 | 16 битов |
| Номер версии прошивки (VersionNumber) | 17 | 16 битов |

1.2. Вычисление значений измеренных величин

Если используются числа с плавающей точкой (UsingFloat=1 только для декодеров T46 и T42), значением измеренной величины является 32-разрядное слово, расположенное в регистрах ввода по адресу:

- 0 – для крутящего момента (силы);
- 2 – для частоты вращения.

Если используются числа с фиксированной точкой (UsingFloat=0 для всех декодеров), значением измеренной величины вычисляется по формуле:

- **Момент/сила** (единица измерения) = $MomentInt * 10^{MomentExp}$;
- **Частота вращения** (rpm) = $RotationFrequencyInt * 10^{RotationFrequencyExp}$;

Формула для вычисления температуры: **Температура** (°C) = $TemperatureInt / 10$;

В обоих случаях единица измерения для крутящего момента (силы) одинакова и зависит только от номинального крутящего момента (силы) датчика и автоматически учитывается декодером. Возможные единицы измерения приведены в таблице.

| Номинальный крутящий момент (сила) | Единица измерения получаемых данных |
|------------------------------------|-------------------------------------|
| 0.1 Nm (N) ÷ 8 Nm (N) | mNm (mN) |
| 10 Nm (N) ÷ 8 kNm (N) | Nm (N) |
| 10 kNm (N) и более | kNm (kN) |

2. Диалоговый режим работы

2.1. Общий формат запросов и ответов

Все запросы и ответы имеют формат:

| | | |
|-----------|--------|-------------------|
| Заголовок | Данные | Контрольная сумма |
|-----------|--------|-------------------|

- Заголовок;
- Данные;
- Контрольная сумма.

2.2. Формат заголовка и контрольной суммы для декодеров T46

Для декодеров T46 заголовки запроса и ответа имеют формат:

| | |
|------------------|-------------|
| Адрес устройства | Код команды |
|------------------|-------------|

- адрес устройства (1 байт) – адрес устройства, к которому адресован запрос (имеет смысл только для декодера T46). Адрес устройства может изменяться от 1 до 247. Адреса в диапазоне 248...255 – зарезервированы.
- код команды (1 байт) – говорит ведомому устройству, какие данные или выполнение какого действия требует от него ведущее устройство. Старшая единица (0x80) в коде команды используется в ответе подчиненного, чтобы указать, что операция закончилась с ошибкой;

Контрольная сумма (2 байта):

| |
|-------------------|
| Контрольная сумма |
|-------------------|

Контрольная сумма служит для проверки отсутствия ошибок в запросе и ответе. Это циклическая контрольная сумма CRC-16 Modbus.

2.3. Формат заголовка и контрольной суммы для декодеров T42, T45 и T47

Для декодеров T45 и T47 заголовки запроса и ответа имеют формат.

Заголовок запроса:

| |
|-------------|
| Код команды |
|-------------|

- код команды (1 байт);

Для декодеров T45 и T47 контрольная сумма – отсутствует.

Для декодеров T42 контрольная сумма присутствует.

2.4. Коды команд декодеров

Все декодеры имеют одинаковый набор команд:

- 3 – READ_HOLDING_REGISTERS – чтение значений из нескольких регистров хранения;
- 4 – READ_INPUT_REGISTERS – чтение значений из нескольких регистров ввода;
- 5 – FORCE_SINGLE_COIL – запись значения одного флага;
- 6 – PRESET_SINGLE_REGISTER – запись значения в один регистр хранения;
- 16 – PRESET_MULTIPLE_REGISTERS – запись значений в несколько регистров хранения;
- 17 – REPORT_SLAVE_ID – чтение служебной информации об устройстве.

2.5. Формат поля данных запросов

- *Формат данных запроса READ_HOLDING_REGISTERS (3)*
 - Адрес первого элемента таблицы регистров хранения (2 байта);
 - Количество считываемых регистров (2 байта).

- **Формат данных запроса READ_INPUT_REGISTERS (4)**
 - Адрес первого элемента таблицы регистров ввода (2 байта);
 - Количество считываемых регистров (2 байта).
- **Формат данных запроса FORCE_SINGLE_COIL (5)**
 - Адрес регистра (2 байта);
 - Устанавливаемое значение (2 байта). Для флагов значение 0xFF00 означает включённое состояние, 0x0000 — выключенное, другие значения недопустимы.
- **Формат данных запроса PRESET_SINGLE_REGISTER (6)**
 - Адрес регистра (2 байта);
 - Устанавливаемое значение (2 байта).
- **Формат данных запроса PRESET_MULTIPLE_REGISTERS (16)**
 - Адрес регистра (2 байта);
 - Количество записываемых регистров (2 байта);
 - Количество передаваемых байт - N (1 байт);
 - Устанавливаемые значения (N байт).
- **Формат данных запроса REPORT_SLAVE_ID (17)**

Поле данных в этом запросе пустое.

2.6. Формат поля данных ответов

Если во время выполнения команды произошла ошибка, то декодер возвращает в поле данных код завершения (1 байт). Код завершения может принимать значения:

- **RET_CODE_COMMAND_ERR (1)** – Принятый код функции не может быть обработан.
- **RET_CODE_ADDRESS_ERR (2)** – Адрес данных, указанный в запросе, недоступен.
- **RET_CODE_DATA_ERR (3)** – Значение, содержащееся в поле данных запроса, является недопустимой величиной.
- **RET_CODE_UNREPAIRABLE_ERR (4)** – Невосстанавливаемая (unrepairable) ошибка имела место, пока ведомое устройство пыталось выполнить затребованное действие.
- **RET_CODE_BUSY_ERR (6)** – Ведомое устройство занято обработкой команды. Ведущее устройство должно повторить сообщение позже, когда ведомое освободится.
- **RET_CODE_CHECKSUM_ERR (8)** – Ведомое устройство обнаружило ошибку паритета.

Признаком того, что во время выполнения команды произошла ошибка, служит 1 в старшем разряде кода команды в заголовке ответа.

Ниже приведены форматы данных ответов в случае успешного завершения операции.

- **Формат данных ответа на команду READ_HOLDING_REGISTERS (3)**
 - Количество передаваемых байт - N (1 байт);
 - Запрошенные регистры (N байт);
- **Формат данных ответа на команду READ_INPUT_REGISTERS (4)**
 - Количество передаваемых байт - N (1 байт);
 - Запрошенные регистры (N байт);
- **Формат данных ответа на команду FORCE_SINGLE_COIL (5)**
 - Адрес регистра (2 байта);
 - Устанавливаемое значение (2 байта). Для флагов значение 0xFF00 означает включённое состояние, 0x0000 — выключенное, другие значения недопустимы.

- **Формат данных ответа на команду PRESET_SINGLE_REGISTER (6)**
 - Адрес регистра (2 байта);
 - Устанавливаемое значение (2 байта).
- **Формат данных ответа на команду PRESET_MULTIPLE_REGISTERS (16)**
 - Адрес регистра (2 байта);
 - Количество записываемых регистров (2 байта);
- **Формат данных ответа на команду REPORT_SLAVE_ID (17)**
 - Идентификатор датчика (3 байта);
 - Температура (целое число 1 байт);
 - Коррекция чувствительности (целое число 1 байт);
 - Количество зубьев (Целое число - 2 байта);
 - Максимальная скорость вращения, деленная на 100 (целое число 1 байт);
 - Дата поверки (3 байта);
 - Дополнительная текстовая информация (49 байт);

2.7. Порядок передачи байтов в словах

В соответствии с протоколом Modbus, декодер T46 принимает запросы и передает ответы, в которых первым байтом всегда передается **старший** байт.

Остальные декодеры (T42, T45 и T47) принимают запросы и передают ответы, в которых первым байтом всегда передается **младший** байт.

2.8. Примеры запросов и ответов для декодера T46

- **Команда FORCE_SINGLE_COIL (5)**

Требуется установить бит конфигурации «Старт/Стоп» в 1, т. е. начать измерения.

Адрес устройства = 1.

Адрес регистра в разделе регистров хранения = 0x0000.

Слово для записи = FF00

Запрос: 01 05 0000 FF00 8C3A

Ответ: 01 05 FF00 5029

- **Команда PRESET_SINGLE_REGISTER (6)**

Требуется установить коэффициент усреднения равным 100.

Адрес устройства = 1.

Адрес регистра в разделе регистров хранения = 0x0001.

Новое значение = 0x0064

Запрос: 01 06 0001 0064 D9E1

Ответ: 01 06 0001 0064 D9E1

- **Команда PRESET_MULTIPLE_REGISTERS (16)**

Требуется установить время на часах декодера равным 0.

Адрес устройства = 1.

Адрес первого регистра в разделе регистров хранения = 0x0003.

Количество регистров = 2.

Новые значения: младшая часть = 0x0000, старшая часть = 0x0000.

Запрос: 01 10 0003 0002 04 0000 0000 B3BA

Ответ: 01 10 0003 0002 B1C8

- **Команда READ_HOLDING_REGISTERS (3)**

Требуется прочитать текущее время на часах декодера.

Адрес устройства = 1.

Адрес первого регистра в разделе регистров хранения = 0x0003.

Количество регистров = 2.

Запрос: 01 03 0003 0002 340B

Ответ: 01 03 04 B0C1 002E 0D13

- **Команда READ_INPUT_REGISTERS (4)**

Требуется прочитать 5 регистров ввода, начиная с адреса 0.

Адрес устройства = 1.

Адрес первого регистра в разделе регистров ввода = 0x0000.

Количество регистров = 2.

Запрос: 01 04 0000 0005 3009

Ответ: 01 04 0A 0FA0 0000 0E4F FFFE 012C 1C03

- **Команда REPORT_SLAVE_ID (17)**

Требуется прочитать служебную информацию датчика.

Адрес устройства = 1.

Запрос: 01 11 C02C

Ответ: 01 11 043500A07F01003201050CD3E0F0E8F0EEE2E0EB20C1F3EBE0E2EAEE20C3
E8F1F2E5F0E57E8F120302C312520D2EEF7EEDEEF1F2FC20302C31352500 245E

3. Поточковый режим работы

Декодеры T42, T45 и T47 имеют возможность работы в режиме потоковой передачи. Назначение этого режима работы – передать весь поток измерений в главный компьютер.

Режим потоковой передачи включается, как только будут установлены в 1 флаги «Старт/Стоп» и «Потоковая передача». При этом декодер будет автоматически формировать буфера из нескольких измерений крутящего момента (силы) и передавать их в главный компьютер. Размер буфера зависит от типа декодера и коэффициента усреднения (AveragingFactor).

Для вращающихся датчиков в соответствии с алгоритмом вычисления частоты вращения, примерно 10 раз в секунду, главному компьютеру будет передаваться текущая частота вращения.

Температура передается в главный компьютер, как только обнаружится ее изменение.

В режиме потоковой передачи крутящий момент (сила) передается только в формате чисел с фиксированной точкой. Положение фиксированной точки одинаковое для для всех измерений и может быть прочитано из регистра ввода «MomentExp». Формула вычисления крутящего момента (силы) такая же, как и для диалогового режима. Частота вращения и температура передаются в формате чисел с плавающей точкой и дополнительных вычислений не требуют.

В декодерах T47 режим потоковой передачи может работать параллельно с диалоговым режимом, так как в этом декодере используется дуплексная линия связи. В декодерах T42 и T45 линия связи полудуплексная, поэтому для организации диалогового режима нужно остановить потоковую передачу или полностью остановить измерения.

3.1. Формат буферов для передачи в главный компьютер

Общий формат буферов:

| | | |
|-----------|--------|-------------------|
| Заголовок | Данные | Контрольная сумма |
|-----------|--------|-------------------|

Контрольная сумма присутствует только у декодеров T42.

Заголовок имеет формат:

| | |
|------------|------------------------|
| Тип данных | Длина данных (2 байта) |
|------------|------------------------|

Где: Тип данных:

- **100 – крутящий момент (сила).** Данные представляют собой структуру:
//----- формат буфера с крутящим моментом (силой)

```
struct _MeasuresBuffer {
    short int TimeLow;           // Время от старта измерений - мл часть
    short int TimeHigh;         // Время от старта измерений - старшая часть
    unsigned char BufferCount;   // номер буфера
    unsigned char MeasuresBuffersCount; // Размер очереди
    short int Moments[DATA_BUFFER_SIZE]; // Измерения
};
```
- **101 – частота вращения.** Данные представляют собой структуру:
//----- формат буфера с частотой вращения

```
struct _RotationFrequency {
    unsigned short int TimeLow;
    unsigned short int TimeHigh;
    float RotationFrequency;
};
```
- **102 – температура.** Данные представляют собой структуру:
//----- формат буфера с температурой

```
struct _TemperaturaBuffer {
    short int TimeLow;
    short int TimeHigh;
    float Temperatura;
};
```



```
};
```

- **103 – сообщения.** Данные представляют собой структуру:
//----- формат буфера с сообщениями
struct _MessageBuffer {
 short int TimeLow;
 short int TimeHigh;
 short int MessageCount;
 short int Messages[MessageCount];
};

4. Приложение 1. Коды сообщений

```
#define MESSAGE_POLUCHEN_ID      2 // Сообщение "получена служебная информация!"
#define MESSAGE_OSHIBKA_ID      3 // Сообщение "служебная информация не совпадает со
старой"
#define MESSAGE_DATCHIK_OFF     4 // Датчик отключился
#define MESSAGE_DATCHIK_ON      5 // Датчик подключился
#define MESSAGE_BUFFER_LOSS     15 // Потеря буфера данных
#define MESSAGE_MESSAGEQ_OVERFL 19 // Была переполнена очередь сообщений
```