

# 1 Лабораторная работа №1. Изучение протокола обмена данными MODBUS-RTU

## 1.1 Общие сведения о протоколе MODBUS-RTU

Протокол MODBUS-RTU позволяет реализовать обмен данными между устройствами в виде чтения и записи значений в регистры. Как правило, он используется для связи с контроллерами по линии RS-485, предусматривающей подключение нескольких устройств к одной линии. При этом каждому устройству присваивается уникальный адрес в диапазоне 0...247. Устройство с адресом 0 называется головным, и всегда является инициатором обмена данными. Устройства с адресами 1...247 являются подчиненными, и большую часть времени ожидают прихода команды от головного устройства. Обмен данными выполняется по следующей схеме.

- Головное устройство отправляет пакет (сообщение), в котором указывается адрес подчиненного устройства, к которому он передается, код функции (чтение или запись регистра) и, при необходимости, данные.
- Все подчиненные устройства принимают этот пакет. Если адрес подчиненного устройства совпал с адресом, указанным в пакете, устройство выполняет необходимую функцию и посылает в головное устройство ответный пакет.
- Головное устройство принимает и обрабатывает ответный пакет (анализирует его на наличие ошибок, копирует необходимые данные).

## 1.2 Передача и проверка контрольной суммы

При обмене данными по протоколу MODBUS-RTU в конце каждого пакета передается контрольная сумма – 16-битное значение, вычисляемое с использованием значений всех байт пакета, не считая самой контрольной суммы. Ее передача необходима для обеспечения высокой достоверности передаваемой информации. При получении пакета с неверной контрольной суммой, устройство не должно использовать содержащиеся в нем данные для дальнейшей работы, т.е. факт получения пакета должен игнорироваться.

Контрольная сумма в протоколе MODBUS-RTU вычисляется как избыточный циклический код с образующим полиномом A001. Вычисление производится по следующему алгоритму:

1. **CRC = FFFF;**
2. с использованием значения очередного байта **X** выполняется операция **CRC=CRC^X;**
3. сохраняется значение младшего бита **CRC: L=CRC&1;**
4. **CRC** сдвигается вправо на 1 бит;
5. если **L=1**, выполняется операция **CRC=CRC^A001;**
6. если выполнено меньше 8 сдвигов **CRC**, происходит переход к п.3;
7. если обработаны не все байты блока данных, происходит переход к п.2;
8. в **CRC** находится контрольная сумма.

На языке C++ алгоритм реализуется следующим образом:

```
unsigned CalcCRC( unsigned char *Buf, unsigned Len ) {
```

```

unsigned CRC = 0xFFFF;
for ( unsigned i=0; i<Len; ++i, ++Buf ) {
    CRC ^= static_cast<unsigned char>(*Buf);
    for ( unsigned j=0; j<8; ++j ) {
        bool LSB = CRC & 0x0001;
        CRC >>= 1;
        if(LSB) CRC ^= 0xA001;
    }
}
return CRC;
}

```

В лабораторной работе для вычисления контрольных сумм может быть использовано приложение ModBusCRCWin.exe.

### 1.3 Функция 3 – чтение состояния регистров

Для чтения состояния регистра головное устройство передает в подчиненное пакет, формат которого приведен в таблице (Таблица 1). Для 16-битных значений в пакете первым должен следовать старший байт, вторым – младший (для значений, хранящихся в памяти компьютера порядок следования байтов обратный).

Таблица 1. Формат пакета чтения состояния регистра (параметры приведены в порядке их следования в пакете)

<i>Параметр</i>	<i>Размер, бит</i>	<i>Значение</i>
Адрес подчиненного устройства	8	1
Код функции	8	3
Адрес регистра	16	–
Количество считываемых регистров	16	–
Контрольная сумма	16	–

Формат ответного пакета, содержащего состояние регистров, приведен в таблице (Таблица 2).

Таблица 2. Формат ответного пакета при чтении состояния регистра

<i>Параметр</i>	<i>Размер, бит</i>	<i>Значение</i>
Адрес подчиненного устройства	8	1
Код функции	8	3
Количество считываемых байт	8	–
Состояние регистра	16	–
Контрольная сумма	16	–

### 1.4 Функция 16 – запись значений в регистры

Форматы пакетов, передаваемых головным и подчиненным устройствами, приведены в таблицах (Таблица 3, Таблица 4).

Таблица 3. Формат пакета записи состояния регистра

<i>Параметр</i>	<i>Размер, бит</i>	<i>Значение</i>
Адрес подчиненного устройства	8	1
Код функции	8	16
Адрес регистра	16	–
Количество записываемых регистров	16	–
Количество записываемых байт	8	–
Записываемое значение	16	–
Контрольная сумма	16	–

Таблица 4. Формат ответного при записи состояния регистра

<i>Параметр</i>	<i>Размер, бит</i>	<i>Значение</i>
Адрес подчиненного устройства	8	1
Код функции	8	16
Адрес регистра	16	–
Количество записанных регистров	16	–
Контрольная сумма	16	–

## **1.5 Программа работы**

1. Сформировать сообщения (пакеты) для записи команд в регистры устройства для варианта, заданного преподавателем. Адреса устройств и регистров заданы в таблицах (Таблица 5, Таблица 6). Значение сигнала K\_int-1 принять равным номеру варианта плюс 16, для сигналов K\_int-2, K\_bool-01...K\_bool-16 использовать произвольные значения.
2. Подключить порт контроллера RAPTOR-A9, соответствующий номеру варианта (Таблица 5), к линии связи и проверить выполнение команд, заданных в пакетах (п.1). Отправка пакетов и просмотр ответов выполняется с помощью терминала (специального приложения Windows). Контроллер настроен на скорость передачи данных 9600 бит/с, бит четности не используется, количество стоповых битов: 1.
3. Сформировать сообщения (пакеты) для чтения ТИ и ТС из регистров устройства. Адреса устройств и регистров заданы в таблицах (Таблица 5, Таблица 7).
4. Проверить выполнение запросов на чтение регистров, заданных в пакетах (п.3). Проверить совпадение принимаемых значений ТИ/ТС со значениями, заданными в графическом интерфейсе контроллера.
5. Сформировать пакеты для чтения регистров измерительного преобразователя РМ130Р или ЭНИП-2, содержащих данные о токах, напряжениях и мощности. Для ЭНИП-2 сформировать также пакет для чтения состояния дискретных входов.

6. Подключить измерительный преобразователь к линии связи и проверить выполнение команд, заданных в пакетах (п.5). В отчете привести состояние считанных регистров.
7. По данным о состоянии регистров, полученным в п.6, рассчитать измеренные измерительным преобразователем напряжения, токи и мощности в именованных единицах.

Таблица 5. Адреса устройств, реализованных в контроллере RAPTOR-A9

<i>№ варианта (№ устройства)</i>	<i>Номер СОМ-порта</i>	<i>Адрес устройства</i>	<i>Начальный адрес ТИ/ТС (Addr_F03)</i>
1	1	5	10
2	2	10	20
3	3	15	30
4	4	20	40
5	5	25	50

Таблица 6. Адреса командных регистров (код функции 16)

<i>Адрес регистра</i>	<i>Параметр</i>
1	Команда K_int-1 (целое число)
2	Команда K_int-2 (целое число)
3	Дискретные команды K_bool-01 ... K_bool-16 (K_bool-01 – младший бит регистра, ... K_bool-16 – старший бит)

Таблица 7. Адреса регистров ТИ/ТС (код функции 3)

<i>Адрес регистра</i>	<i>Параметр</i>
<i>Addr_F03</i>	ТИ-1 (целое число)
<i>Addr_F03 + 1</i>	ТИ-2 (целое число)
<i>Addr_F03 + 2</i>	ТС-1 (младший/первый бит), ТС-2 (второй бит)