

Протокол связи преобразователя интерфейсного Ethernet/RS-485 (версия 8)

Дата последнего изменения: 10.12.2012

Оглавление

1. Описание устройства	2
2. Параметры настройки по умолчанию.....	2
3. Поиск преобразователя с использованием широковещательного запроса	2
4. «Прозрачный» режим	3
5. Конфигурирование преобразователя через интерфейс RS-485.....	3
6. Формат пакетов в командном режиме.....	3
6.1 Формат пакета конфигурирования.....	4
6.2 Команды конфигурирования преобразователя	4
6.2.1 Команда СБРОС.....	4
6.2.2 Команда чтения версии прошивки и сетевых настроек	4
6.2.3 Команда изменения сетевых настроек	5
6.2.4 Команда чтения настроек «прозрачного» режима	7
6.2.5 Команда записи настроек «прозрачного» режима	7
6.2.6 Команда смены пароля	8
6.3 Формат пакета ретрансляции данных TCP -> RS-485	9
7. Приложение А: определение значений управляющих полей SP и BRGH, исходя из требуемой скорости связи по интерфейсу RS-485	12
8. Приложение Б: подпрограмма расчета контрольной суммы.....	13

1. Общие сведения

Преобразователь интерфейсный Ethernet/RS-485 (в дальнейшем преобразователь) предназначен для подключения устройств, обладающих интерфейсом RS-485, к компьютеру через TCP соединение.

Поддерживаются 2 режима работы:

- режим «прозрачный» (подключение через TCP сокет, при этом все получаемые по сети байты транслируются на интерфейс RS-485, а все байты, получаемые по RS-485, передаются на TCP-сокет),
- режим «запрос-ответ» с подтверждением передачи данных.

Преобразователь позволяет изменять параметры как через Ethernet соединение, так и через RS-485. При этом для предотвращения несанкционированного изменения настроек через Ethernet есть возможность установить 8-символьный пароль. Изменение параметров через RS-485 соединение не требует пароля.

Преобразователь поддерживает протокол ICMP, что позволяет проверять связь посредством утилиты ping.

Преобразователь поддерживает функцию поиска через широковещательный запрос.

2. Параметры настройки по умолчанию

- Сетевые параметры:
 - IP адрес 192.168.0.5
 - Шлюз 192.168.0.1
 - Маска подсети 255.255.255.0
 - Командный TCP порт 1000
 - Управление потоком Ethernet полудуплекс
- Параметры "прозрачного" режима:
 - Порт TCP 1001
 - Скорость передачи данных по RS-485..... 9600 бод
 - Тайм-аут сборки пакетов 0
 - Число бит данных 8
 - Число стоп бит..... 1
- Пароль для изменения настроек модема через TCP соединение
<00><00><00><00><00><00><00><00> (8 нулевых байт).

3. Поиск преобразователя с использованием широковещательного запроса

Преобразователь принимает широковещательные UDP пакеты на порт 30303. Если в первом байте данных есть символ "D", то на IP адрес отправителя пакета будет отправлен UDP пакет,

содержащий IP адрес и командный порт в виде HEX строки. Порядок байт сетевой, разделитель "_".

Например, для адреса 192.168.0.5:1000 будет передана строка "C0A80005_03E8"

4. «Прозрачный» режим

Режим прямого обмена данными без формирования каких-либо заголовков и контрольных сумм называется "прозрачным". Все данные, приходящие на TCP порт "прозрачного" режима, отправляются на порт RS-485 и наоборот все данные, полученные с порта RS-485, будут отправлены на TCP порт "прозрачного" режима.

Примечание: Преобразователь поддерживает одновременно только одно TCP соединение. То есть, чтобы изменить настройки и подключиться к командному порту, надо закрыть соединение на порт «прозрачного» режима и наоборот.

5. Конфигурирование преобразователя через интерфейс RS-485

Переход в режим конфигурирования через интерфейс RS-485 выполняется по включению питания. Время ожидания команд составляет 1,5 с, после чего преобразователь переходит в основной режим работы.

Если за это время получена команда с корректной контрольной суммой, то преобразователь остаётся в режиме ожидания команд ещё на 1,5 с. Если за это время новая команда не получена или получены случайные байты, то преобразователь переходит в основной режим работы (ожидает подключения по Ethernet, игнорирует любую передачу по RS-485). При получении команды СБРОС через RS-485 преобразователь сразу же переходит в основной режим работы.

При перезагрузке по команде СБРОС, переданной через Ethernet, преобразователь сразу переходит в основной режим работы без задержки в 1,5 с.

Формат команд, передаваемых через RS-485, полностью совпадает с форматом команд, передаваемых через командный порт Ethernet, за исключением того, что изменение настроек не требует указания поля пароля. Таким образом, если пароль утерян, его можно сбросить с помощью стандартной утилиты настройки через соединение RS-485.

Команды в режиме конфигурирования через RS-485 передаются на скорости 9600 бит/с, 8 бит данных, 1 бит стоповый.

6. Формат пакетов в командном режиме

Пакетная передача используется при соединении через командный TCP порт и в режиме конфигурирования через RS-485. Через RS-485 доступны только пакеты конфигурирования, в то время как через командный TCP порт дополнительно присутствует команда ретрансляции данных.

6.1 Формат пакета конфигурирования

Данный тип пакетов позволяет прочесть информацию о версии прошивки, сервисную информацию, прочесть и изменить текущие параметры работы преобразователя.

При работе через TCP соединение все команды, изменяющие параметры работы преобразователя, требуют указания пароля.

Формат пакета запроса:

<N> <PU> <CMD> <DD₁>...<DD_N> <CRC16_L> <CRC16_H>

Все поля имеют длину 1 байт.

- N - длина информационной части пакета (поле DD) в байтах. Может принимать значения в диапазоне 0 - 255.
- PU – битовое поле управления. Для команд конфигурации всегда равно 0x80
- CMD - код команды для преобразователя.
- DD₁..DD_N – N байт дополнительных данных команды.
- <CRC16_L> <CRC16_H> – контрольная сумма всех остальных байт пакета ([см. приложение Б](#))

Формат пакета ответа:

<N> <PU> <CMD> <DD₁>...<DD_N> <CRC16_L> <CRC16_H>

Все поля имеют длину 1 байт.

- N - длина информационной части пакета (поле DD) в байтах. Может принимать значения в диапазоне 0 - 255.
- PU – битовое поле управления. Копируется значение поля из пакета запроса.
- CMD - код команды для преобразователя. Копируется значение поля из пакета запроса.
- DD₁..DD_N – N байт дополнительных данных команды.
- <CRC16_L> <CRC16_H> – контрольная сумма всех остальных полей пакета ([см. приложение Б](#))

6.2 Команды конфигурирования преобразователя

6.2.1 Команда СБРОС

Код команды: **0x20**

Дополнительные данные в пакете: <нет>

Ответ: Преобразователь выполняет перезапуск программы, ответ на команду не возвращает.

Команда может использоваться для инициирования вступления в силу измененных сетевых настроек преобразователя.

Пример запроса:

00 80 20 CA 24

6.2.2 Команда чтения версии прошивки и сетевых настроек

Код команды: **0x40**

Дополнительные данные в пакете: <нет>

Ответ: 30 байт:

- версия прошивки – 1байт;
- MAC адрес – 6 байт, сетевой порядок следования байт;
- IP адрес – 4 байта, сетевой порядок следования байт;
- IP шлюза – 4 байта, сетевой порядок следования байт;
- Командный порт – 2 байта, младшим байтом вперёд;
- маска подсети – 4 байта, сетевой порядок следования байт;
- SystemClock – 4 байта, младшим байтом вперёд - частота микропроцессора в Гц;
- режим работы Ethernet контроллера – 1 байт:
 - 0 – полудуплекс,
 - 1 – дуплекс;
- IP адрес клиента при последнем соединении – 4 байта. Используется для обнаружения несанкционированных подключений. При первом соединении после включения питания это поле возвращается равным 0.0.0.0. При перезагрузке по [команде СБРОС](#) адрес последнего соединения не сбрасывается. При соединении на командный порт или порт «прозрачного» режима адрес клиента сохраняется, чтобы быть переданным при следующем соединении.

Пример запроса:

00 80 40 8A 25

Пример ответа:

1E 80 40 08 02 01 02 03 04 05 C0 A8 00 05 C0 A8 00 01 E8 03 FF FF FF 00 40 78 7D 01 00 00 00 00 62 70

Расшифровка ответа:

- версия прошивки..... 8
- MAC адрес..... 02:01:02:03:04:05
- IP адрес 192.168.0.5
- IP шлюза..... 192.168.0.1
- командный порт 1000
- маска адреса..... 255.255.255.0
- частота микропроцессора..... 25МГц
- режим работы Ethernet контроллера полудуплекс
- IP адрес клиента при последнем соединении – текущее соединение является первым после загрузки преобразователя.

6.2.3 Команда изменения сетевых настроек

Код команды: **0x60**

Дополнительные данные в пакете - 29 байт:

- пароль для изменения настроек¹ – 8 байт (дополняется до восьми байт нулями);
- MAC адрес – 6 байт, сетевой порядок следования байт;
- IP адрес – 4 байта, сетевой порядок следования байт;
- IP шлюза – 4 байта, сетевой порядок следования байт;
- командный порт – 2 байта, младшим байтом вперёд;
- маска подсети – 4 байта, сетевой порядок следования байт;
- режим работы Ethernet контроллера:
 - 0 – полудуплекс,
 - 1 – дуплекс/

Ответ 1 байт – код ошибки (если есть):

- 0x00 – команда выполнена успешно;
- 0x81 – ошибка проверки пароля (обработка команд изменения параметров блокируется на 10 с);
- 0x82 – изменение параметров заблокировано на 10 секунд после передачи неверного пароля;
- 0x83 – неверная длина команды, или какой-либо из параметров имеет недопустимое значение.

Пример запроса через командный порт Ethernet:

1D 80 60 00 00 00 00 00 00 00 00 02 01 02 03 04 05 C0 A8 00 05 C0 A8 00 01 E8 03 FF FF FF 00 00
84 DE

Расшифровка запроса:

- Поле пароля..... 00 00 00 00 00 00 00 00 (пустой пароль);
- MAC адрес..... 02:01:02:03:04:05;
- IP адрес 192.168.0.5;
- IP шлюза..... 192.168.0.1;
- командный порт 1000;
- маска подсети..... 255.255.255.0;
- режим работы Ethernet контроллера.... 0 – полудуплекс.

Тот же запрос при конфигурировании через интерфейс RS485:

15 80 60 02 01 02 03 04 05 C0 A8 00 05 C0 A8 00 01 E8 03 FF FF FF 00 00 87 97

Пример ответа:

01 80 60 00 DD DE

Расшифровка ответа: параметры записаны корректно

Примечание: новые параметры вступят в силу при следующей загрузке преобразователя. Для этого используется [команда СБРОС](#).

¹ В режиме конфигурирования через интерфейс RS-485 это поле опускается.

6.2.4 Команда чтения настроек «прозрачного» режима

Код команды: **0x11**.

Дополнительные данные в пакете: <нет>.

Ответ 6 байт:

- порт «прозрачного» режима – 2 байта, младшим байтом вперёд (если равен 0, то «прозрачный» режим отключен);
- SP – 1 байт – делитель частоты, вместе с битом BRGH (передается в поле Cfg.0) определяет скорость обмена через RS-485 в «прозрачном» режиме ([см. приложение А](#));
- тайм-аут сборки пакетов, мс – 2 байта, младшим байтом вперёд - при получении очередного байта по RS-485 будет выждан указанный тайм-аут до отправки данных через Ethernet. Если в течение этого времени придет очередной байт – тайм-аут начнёт отсчитываться заново. Если принято 255 байт, то они в любом случае будут немедленно отправлены. Если тайм-аут равен 0, данные отправляются сразу по получении.
- Cfg – 1 байт – битовый регистр дополнительных настроек
 - Бит Cfg.2 – 0 – 8-битовая передача, 1 – 9-битовая
 - Бит Cfg.1 – если используется 9-битовая передача, то задаёт значение 9 бита
 - Бит Cfg.0 – бит BRGH, задаёт режим высокой скорости обмена через RS-485 ([см. приложение А](#))
 Остальные биты зарезервированы.

Пример запроса:

00 80 11 6F A4

Пример ответа:

06 80 11 E9 03 A2 02 00 01 35 76

Расшифровка ответа:

- Порт «прозрачного» режима 1001
 - Тайм-аут сборки пакетов..... 2 мс
 - SP..... 162
 - Cfg..... 01 (8-битовая передача, высокая скорость передачи)
- То есть скорость передачи 9600 бит/с ([см. пример 2 в приложении А](#))

6.2.5 Команда записи настроек «прозрачного» режима

Код команды: **0x12**

Дополнительные данные в пакете: 20 байт

- Пароль для изменения настроек² – 8 байт (дополняется до восьми байт нулями)
- Порт «прозрачного» режима – 2 байта, младшим байтом вперёд (если равен 0, то «прозрачный» режим отключен)
- SP - 1 байт – делитель частоты, вместе с битом BRGH (передается в поле Cfg.0) определяет скорость обмена через RS-485 в «прозрачном» режиме ([см. приложение А](#))

² В режиме конфигурирования через интерфейс RS485 это поле опускается.

- Тайм-аут сборки пакетов, мс – 2 байта, младшим байтом вперёд - при получении очередного байта по RS-485 будет выждан указанный тайм-аут до отправки данных через Ethernet. Если в течение этого времени придет очередной байт – тайм-аут начнёт отсчитываться заново. Если принято 255 байт, то они в любом случае будут немедленно отправлены. Если тайм-аут равен 0, данные отправляются сразу по получении.
- Cfg - 1 байт – битовый регистр дополнительных настроек
 - Бит Cfg.2 - если = 0 – 8-битовая передача
если = 1 – 9-битовая передача
 - Бит Cfg.1 - если используется 9-битовая передача, то задаёт значение 9 бита
 - Бит Cfg.0 - бит BRGH, задаёт режим высокой скорости обмена через RS-485 ([см. приложение А](#))
Остальные биты зарезервированы и должны быть равны 0.

Ответ: 1 байт – код ошибки (если есть)

- 0x00 – команда выполнена успешно
- 0x81 – ошибка проверки пароля (обработка команд изменения параметров блокируется на 10 с)
- 0x82 – изменение параметров заблокировано на 10 секунд после передачи неверного пароля
- 0x83 – неверная длина команды или какой-либо из параметров имеет недопустимое значение.

Примечание: новые параметры вступят в силу при следующей загрузке преобразователя. Для этого используется [команда СБРОС](#).

Пример запроса:

0E 80 12 00 00 00 00 00 00 00 00 E9 03 A2 02 00 01 01 2D

Расшифровка запроса:

- Поле пароля.....00 00 00 00 00 00 00 00 (пустой пароль)
- Порт «прозрачного» режима1001
- Тайм-аут сборки пакетов..... 2 мс
- SP..... 162
- Cfg.....01 (8-битовая передача, высокая скорость передачи)

То есть скорость передачи 9600 бит/с ([см. пример 2 в приложении А](#)).

Тот же запрос при конфигурировании через интерфейс RS-485:

06 80 12 E9 03 A2 02 00 01 05 75

Пример ответа:

01 80 12 00 D8 72

Расшифровка ответа: параметры записаны корректно

6.2.6 Команда смены пароля

Код команды: **0x13**

Дополнительные данные в пакете: 16 байт

- Текущий пароль для изменения настроек³ – 8 байт (дополняется до восьми байт нулями)
- Новый пароль для изменения настроек – 8 байт (дополняется до восьми байт нулями)

Ответ: 1 байт – код ошибки (если есть)

- 0x00 – команда выполнена успешно
- 0x81 – ошибка проверки пароля (обработка команд изменения параметров блокируется на 10 с)
- 0x82 – изменение параметров заблокировано на 10 секунд после передачи неверного пароля
- 0x83 – неверная длина команды.

Примечание: пароль может содержать не только буквы, цифры и знаки, но и символы с кодом от 0 до 255. Однако стандартная программа настройки позволяет вводить только символы, которые можно ввести с клавиатуры.

Пример запроса:

10 80 13 00 00 00 00 00 00 00 00 31 32 33 34 35 36 00 00 73 D9

Расшифровка запроса:

- Текущий пароль для изменения настроек – "" (пароль отсутствует)
- Новый пароль для изменения настроек – "123456"

Тот же запрос при конфигурировании через интерфейс RS-485:

08 80 13 31 32 33 34 35 36 00 00 B9 AB

Пример ответа:

01 80 13 00 DB F4

Расшифровка ответа: параметры записаны корректно

6.3 Формат пакета ретрансляции данных TCP -> RS-485

Этот тип пакета используется для обмена данными с устройством, подключенным к порту RS-485 преобразователя. На каждый такой запрос преобразователь возвращает ответ (либо данные ответа подключенного устройства, либо информация о том, что ответ не был получен). Каждый пакет этого типа несёт индивидуальные настройки скорости передачи и тайм-аут ожидания ответа.

Формат пакета запроса:

<N> <PU> <SP> <TOFB16> <TOI16> <WN> <DD₁>...<DD_N> <CRC16_L> <CRC16_H>

Все поля имеют длину 1 байт.

- N - длина информационной части пакета (поле DD) в байтах. Может принимать значения 0 - 255.
- PU - битовое поле управления:
 - Бит PU.7 - идентификатор типа пакета. Для команд ретрансляции равно 0.

³ В режиме конфигурирования через интерфейс RS-485 это поле опускается.

- Бит PU.5 - сброс устройства, подключенного к шине RS-485 (только для устройств, поддерживающих сброс длительным удержанием линий данных в состоянии логического «0»). Если бит равен «1», то перед отправкой данных на устройство на шине RS-485 сформируется логический "0" на 1 с, затем линии переводятся в пассивное состояние на 200 мс. Затем передаются данные из пакета.
- Бит PU.2 - если = 0 – 8-битовая передача
если = 1 – 9-битовая передача
- Бит PU.1 - если используется 9-ти битовая передача, то задаёт значение 9 бита
- Бит PU.0 - бит BRGH задаёт режим высокой скорости обмена через RS-485 ([см. приложение А](#)).

Остальные биты зарезервированы и должны быть равны 0.

- SP - 1 байт - делитель частоты, вместе с битом BRGH (передается в поле PU.0) определяет скорость передачи этого сообщения и ожидаемого ответа через RS-485 ([см. приложение А](#))
- TOFB16 - 2 байта младшим вперёд – тайм-аут ожидания первого байта ответа устройства RS-485 в мс.

Исходя из особенностей реализации, тайм-аут отсчитывается от момента начала передачи по RS-485, то есть он должен включать время передачи данных $DD_{D1}... DD_N$.

Если TOFB16 равен 0, то ждать бесконечно. Если ответ от устройства, подключенного к преобразователю не последует, то, чтобы отдать следующую команду преобразователю придётся разорвать соединение и открыть новое.

- TOI16 – 2 байта младшим вперёд – межбайтовый тайм-аут при приёме ответа устройства RS-485 в мс.

Исходя из особенностей аппаратной реализации, тайм-аут отсчитывается от момента окончания приёма предыдущего байта и заканчивается окончанием приёма последующего байта. Таким образом, значение должно быть больше длительности передачи 1 байта.

Если TOI16 равен 0, то ждать бесконечно. Если ответ от устройства, подключенного к преобразователю не последует, то, чтобы отдать следующую команду преобразователю придётся разорвать соединение и открыть новое.

- WN - 1 байт - количество ожидаемых байт ответа устройства RS-485. Если это поле будет равно 0, преобразователь не будет ждать ответ устройства и вернет ответ с полем N равным 0.
- $DD_{D1}... DD_N$ - данные, которые будут переданы по шине RS-485.
- <CRC16_L> <CRC16_H> – контрольная сумма всех байт пакета кроме байт контрольной суммы ([см. приложение Б](#))

Примечание: преобразователь отправляет переданные данные $DD_{D1}... DD_N$ через RS-485 и ожидает ответ. При достижении любого из далее перечисленных условий преобразователь отправляет по сети Ethernet ответ со всеми полученными данными. Условия окончания ожидания ответа:

- получение WN байт

- вышел тайм-аут получения первого байта(ТОFB16), и не принято ни одного байта
- вышел межбайтовый тайм-аут (ТОI16) после получения любого байта по RS-485 при условии, что за это время не получено ни одного нового байта.

Формат пакета ответа:

<N> <PU> <DD₁>...<DD_N> <CRC16_L> <CRC16_H>

Все поля имеют длину 1 байт.

- N - длина информационной части пакета (поле DD) в байтах. Может принимать значения 0 - 255.
- PU - битовое поле управления. Копируется значение поля из пакета запроса.
- DD₁..DD_N – данные с устройства, подключенного к шине RS-485
- <CRC16_L> <CRC16_H> - контрольная сумма всех байт пакета кроме байт контрольной суммы ([см. приложение Б](#))

Пример запроса:

04 01 A2 64 00 02 00 02 01 02 03 04 78 58

Расшифровка запроса:

- PU01 (8-битовая передача, высокая скорость передачи)
- SP162
- ТОFB16 (тайм-аут ожидания первого байта) 100 мс
- ТОI16 (межбайтовый тайм-аут)2 мс
- WN (количество ожидаемых байт ответа) 2
- Байты для ретрансляции по RS-48501 02 03 04

Учитывая значения SP и PU.0, скорость передачи составит 9600 бит/с ([см. пример 2 в приложении А](#)).

Пример ответа:

02 01 05 06 D8 36

Расшифровка ответа

- Принятый от подключенного к преобразователю устройства ответ 05 06

7. Приложение А: Определение значений управляющих полей SP и BRGH исходя из требуемой скорости связи по интерфейсу RS-485

Значения поля SP и бита BRGH, используемые в командах [чтения](#) и [записи](#) настроек прозрачного режима, а также [команде ретрансляции](#) определяются исходя из необходимой скорости передачи данных (BaudRate) по RS-485 по формуле

$$SP = \text{SystemClock}[\text{Гц}] / \text{BaudRate}[\text{бит/с}] / 16 - 1, \text{ если BRGH} = 1$$

$$SP = \text{SystemClock}[\text{Гц}] / \text{BaudRate}[\text{бит/с}] / 64 - 1, \text{ если BRGH} = 0$$

SystemClock – определяется с помощью команды [чтения версии прошивки и сетевых настроек](#).

Значение бита BRGH выбирается таким, чтобы SP был максимален, но лежал в диапазоне [0 - 255]

Скорость обмена определяется при известном значении поля SP и бита BRGH по формуле:

$$\text{BaudRate}[\text{бит/с}] = \text{SystemClock}[\text{Гц}] / (SP + 1) / 16, \text{ если BRGH} = 1$$

$$\text{BaudRate}[\text{бит/с}] = \text{SystemClock}[\text{Гц}] / (SP + 1) / 64, \text{ если BRGH} = 0$$

SystemClock – определяется с помощью команды [чтения версии прошивки и сетевых настроек](#).

Пример 1: Если требуется скорость передачи данных 9600 бит/с и при чтении пакета версии прошивки была определена частота работы микропроцессора SystemClock = 25 000 000 Гц то:

$$\text{при BRGH} = 1 \text{ получим } SP = \text{SystemClock}[\text{Гц}] / \text{BaudRate}[\text{бит/с}] / 16 - 1 = 161,76 \approx 162$$

$$\text{при BRGH} = 0 \text{ получим } SP = \text{SystemClock}[\text{Гц}] / \text{BaudRate}[\text{бит/с}] / 64 - 1 = 39,69 \approx 40$$

Выбираем значение Cfg.0 так, чтобы значение SP было максимальным, но ≤ 255 , следовательно:

$$\text{BRGH} = 1, SP = 162$$

Пример 2: Если были прочтены значения BRGH = 1 и SP = 162 и при чтении пакета версии прошивки была определена частота работы микропроцессора SystemClock = 25 000 000 Гц, то реальная скорость передачи данных будет:

$$\text{BaudRate}[\text{бит/с}] = \text{SystemClock}[\text{Гц}] / (SP + 1) / 16 = 25\text{e}6 / (162 + 1) / 16 \approx 9585,89$$

9585,89 – это ближайшая к 9600 реализуемая скорость передачи данных.

Погрешность задания скорости составляет:

$$(9600 - 9585,89) / 9600 \approx 0,00147 \text{ или } 0.147\%$$

8. Приложение Б: Подпрограмма расчета контрольной суммы

```

// подпрограмма расчета контрольной суммы приводится на языке C++
// Входные параметры:
// buffer - ссылка на буфер с данными для расчёта CRC
// count - количество байт данных в буфере
// Функция возвращает CRC16 для байт переданного буфера
unsigned short CRC16Gen(unsigned char* buffer, int count)
{
    unsigned int regCRC = 0xFFFF00;
    unsigned int crcPolynom = 0x8005;
    unsigned int polynomCRC = crcPolynom << 8;
    unsigned int crc = 0;
    for (int i = 0; i < count; i++)
    {
        regCRC |= buffer[i];
        for (int j = 0; j < 8; j++)
        {
            regCRC <<= 1;
            if (regCRC & 0x1000000 != 0)
                regCRC ^= polynomCRC;
        }
    }
    // добавляем два нулевых байта
    for (int j = 0; j < 16; j++)
    {
        regCRC <<= 1;
        if (regCRC & 0x1000000 != 0)
            regCRC ^= polynomCRC;
    }
    return (unsigned short) (regCRC >> 8);
}

```