**ПРИЛОЖЕНИЕ Г**

(обязательное)

**ПРОТОКОЛ ОБМЕНА ПРЕОБРАЗОВАТЕЛЯ СН3020 С СЕРВЕРОМ**

Оглавление

[Интерфейс 2](#_Toc445746696)

[Сетевой режим стыка 2](#_Toc445746697)

[Информационный обмен в протоколе ModBus 2](#_Toc445746698)

[Два режима последовательной передачи 3](#_Toc445746699)

[Содержание сообщения MODBUS 3](#_Toc445746700)

[Номенклатура выходных параметров 3](#_Toc445746701)

[Формат float представления выходных значений параметров 3](#_Toc445746702)

[Фиксация (срез) значений 3](#_Toc445746703)

[Чтение кода АЦП 3](#_Toc445746704)

[Рестарт интерфейсного модуля преобразователя СН3020 3](#_Toc445746705)

[Регистр состояния преобразователя СН3020 3](#_Toc445746706)

[Функции контроля и обработки данных 3](#_Toc445746707)

[ГЕНЕРАЦИЯ LRC/CRC 3](#_Toc445746708)

### Интерфейс

Преобразователь СН3020 оснащен двумя независимыми изолированными портами (стыками) связи типа RS‑485: RS485-1 и RS485-2. Каждый из стыков может быть использован в одном из двух режимов:

* сетевой - для связи с Сервером;
* активный - для циклической выдачи результатов измерения на выносные цифровые табло.

Выбор режима работы стыка осуществляется при параметризации преобразователя СН3020.

Через любой из стыков преобразователя СН3020 осуществляется доступ к внутренним переменным и выходным значениям преобразователя СН3020, что используется для проведения операций регулировки, поверки и эксплуатации преобразователя СН3020.

### Сетевой режим стыка

Информационный обмен преобразователя СН3020 с Сервером осуществляется в пакетном режиме по принципу «запрос-ответ» - в протоколе Modbus, FT1.2 или «циклическая выдача» в протоколе ModBus.

В качестве физической среды передачи информации используется двухпроводная магистраль RS‑485 со следующими параметрами:

1. скорость передачи – устанавливается параметром от 110 до 115,2 кбит/с;
2. способ представления информации - побайтовый.

Каждое сообщение состоит из нескольких полей, передающихся друг за другом без разрывов во времени.

Приём преобразователем СН3020 кадров сервера осуществляется асинхронно и параллельно с внутренними циклами измерения и самодиагностики. Интенсивность запросов сервера не влияет на внутреннюю работу преобразователей СН3020.

### Информационный обмен в протоколе ModBus

Преобразователь СН3020 в составе системы является ведомым (режим «по запросу») или ведущим (режим «циклически»), т.е. ведомый не может передавать информацию в канал без запроса ведущего – Сервера, ведущий – передает в канал циклически.

В режиме «по запросу» Сервер посылает адресные запросы преобразователям СН3020 в виде посылок, на что адресованный преобразователь СН3020 посылает ответ в виде последовательности байт. Число байт запроса и ответа не является постоянной величиной и зависит от характера запроса и состояния преобразователя СН3020. Сервер может выдать адресное сообщение отдельному преобразователю или может инициировать широкую передачу сообщения на все преобразователи. Преобразователь СН3020 возвращает сообщение в ответ на запрос, адресуемый именно ему. При широковещательном запросе преобразователи не выдают ответа.

Запрос Сервера или ответ преобразователя СН3020 на запрос не могут быть посланы до истечения интервала спокойного состояния линии после завершения передачи предыдущего сообщения. Максимальное время задержки ответа преобразователя СН3020 на запрос Сервера - 20 мс.

Каждое сообщение начинается с байта адреса и заканчивается двумя байтами контрольной суммы CRC.

Преобразователь СН3020 может не отвечать по одной из причин:

- адрес сообщения не совпал с адресом преобразователя СН3020;

- контрольная сумма (CRC) не совпала с рассчитанной преобразователем СН3020 контрольной суммой;

- обращение Сервера по широковещательному адресу;

- нарушение формата сообщения-запроса.

Формат сообщения запрос-ответ преобразователя СН3020 приведен в таблице Г.1:

Таблица Г.1

|  |  |
| --- | --- |
| Запрос Сервера | Ответ преобразователя СН3020 |
|  |  |
| Адрес преобразователя СН3020 | Адрес преобразователя СН3020 |
| Код функции | Код функции |
| 8 - битные байты данных | 8 - битные байты данных |
| Контрольная сумма (CRC) | Контрольная сумма (CRC) |

Запрос: Код функции в запросе определяет преобразователю, какое действие необходимо произвести. Байты данных запроса содержат информацию, необходимую для выполнения запрошенной функции. Например, код функции 4 подразумевает запрос на чтение содержимого регистров преобразователя СН3020.

Ответ: Если преобразователь СН3020 выдает ответное сообщение, код функции в ответе повторяет код функции в запросе. В байтах данных содержится затребованная информация. Если имеет место ошибка, то код функции модифицируется, и в байтах данных передается причина ошибки.

В режиме «циклически» преобразователь СН3020 с периодом 1 с, выдает в канал измеренные данные. В этом режиме данные выдаются всегда в одной и той же последовательности не зависимо от назначения параметров (см. раздел «Функция выдачи на табло»). Ответ на эту посылку не требуется, т.к. используется широковещательный запрос для выдачи данных.

### Два режима последовательной передачи

Для работы с преобразователями СН3020 может быть использован один из двух способов передачи: ASCII или RTU. Пользователь выбирает необходимый режим вместе с другими параметрами (скорость передачи, режим паритета и т.д.) во время конфигурации каждого преобразователя СН3020.

#### Режим ASCII

При использовании ASCII - режима каждый байт сообщения передается как два ASCII символа. Старшая тетрада байта передается первым символом, младшая – вторым. Главное преимущество этого способа: временной интервал между соседними символами может достигать 1 с без возникновения ошибок при передаче.

#### Формат каждого байта в ASCII-режиме:

Система кодировки: Шестнадцатеричная, ASCII-символы 0-9, A-F

Назначение битов: 1 старт бит

7 бит данных, младшим битом вперед

1 бит паритета; нет бита паритета

1 стоп бит, если есть паритет; 2 бита если нет паритета;

Контрольная сумма: (LRC)

При использовании ASCII-режима каждый байт сообщения содержит два 4-х битных шестнадцатеричных числа.

Каждое сообщение передается непрерывным потоком.

#### Формат каждого байта в RTU-режиме:

Система кодировки: 8-ми битовая двоичная

Назначение битов: 1 старт бит

8 бит данных, младшим значащим разрядом вперед

1 бит паритета; нет бита паритета

1 стоп бит, если есть паритет; 2 бита если нет паритета;

Контрольная сумма: (CRC)

### Содержание сообщения MODBUS

#### ASCII фрейм

В ASCII-режиме сообщение начинается с символа «:» (двоеточие, ASCII 3A hex), и заканчивается последовательностью «возврат каретки-перевод строки» (CR-LF, ASCII 0D и 0A hex).

Допустимые символы для передачи - это шестнадцатеричные цифры 0-9, A-F. Монитор сетевого устройства в сети непрерывно отслеживает символ «двоеточие». Когда он принят, каждое устройство декодирует следующие поле сообщения (поле адреса) и т.д.

Интервалы между символами сообщения могут быть до 1 с. Если интервал больше, то принимающее устройство распознает это как ошибку. Типичный ASCII-фрейм сообщения показан в таблице Г2:

Таблица Г.2

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Назначение: | Старт | Адрес | Функция | Данные | LRC | Конец |
| Длина поля: | 1 символ:  «:» | 2 символа | 2 символа | n символов | 2 символа | 2 символа: CR-LF |

#### RTU фрейм

Байты в сообщении должны идти друг за другом, без разрывов во времени, т.е. за стоповым битом предыдущего байта должен следовать стартовый бит следующего байта, если он есть. Критерием окончания сообщения является гарантированный интервал спокойного состояния линии, длительность которого определяется установленной скоростью обмена для различных способов передачи данных.

В режиме сообщение начинается с интервала спокойного состояния линии, равного времени передачи не менее 3,5 символов при данной скорости передачи в сети. Первым полем передается адрес устройства.

Вслед за последним передаваемым символом также следует интервал спокойного состояния линии продолжительностью передачи не менее 3,5 символов. Новое сообщение может начинаться только после истечения этого интервала.

Фрейм сообщения передается непрерывно. Если интервал спокойного состояния линии продолжительностью передачи 1,5 символа возник во время передачи фрейма, принимающее устройство заканчивает прием сообщения и следующий байт будет воспринят как начало следующего сообщения.

Таким образом, если новое сообщение начнет передаваться до истечения интервала спокойного состояния линии, принимающее устройство может воспринять его как продолжение предыдущего сообщения. В этом случае устанавливается ошибка, так как будет несовпадение контрольных сумм. Типичный фрейм-сообщение показано в таблице Г.3:

Таблица Г.3

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Старт | Адрес | Функция | Данные | СRC | Конец |
| Т1-Т2-Т3-Т4 | 8 бит | 8 бит | n х бит | 16 бит | Т1-Т2-Т3-Т4 |
| Примечание – Т1…Т4 время передачи одного символа. | | | | | |

#### Содержание адресного поля

Адресное поле фрейма содержит два символа (ASCII) или 8 бит (RTU). Допустимый адрес передачи находится в диапазоне 0 - 247. Каждому подчиненному устройству присваивается индивидуальный адрес в пределах от 1 до 247. Если к Серверу подключено несколько преобразователей СН3020, то каждый из преобразователей СН3020 должен иметь уникальный адрес.

Адрес 255 – универсальный (общий) адрес преобразователя СН3020. При обращении к преобразователю по этому адресу преобразователь СН3020 выдает ответное сообщение, независимо от установленного индивидуального адреса. При использовании адреса 255 к Серверу может быть подключен только один преобразователь СН3020.

Адрес 0 используется для широковещательной передачи, его распознает каждое устройство. На запрос по широковещательному адресу преобразователь СН3020 не отвечает. Преобразователь СН3020 по широковещательному адресу обрабатывает только функцию фиксации срезов.

#### Содержание поля функции

Поле функции фрейма содержит два символа (ASCII) или 8 бит (RTU). Диапазон числа 1 -255. Набор функций, поддерживаемых преобразователем СН3020, представлен ниже.

Когда преобразователь СН3020 отвечает Серверу, он использует поле кода функции для фиксации ошибки. В случае нормального ответа преобразователь СН3020 повторяет оригинальный код функции. Если имеет место ошибка, с ответным сообщением преобразователя СН3020 возвращается код функции с установленным в единицу старшим битом.

Например, сообщение от Сервера к преобразователю прочитать группу регистров имеет следующий код функции:

0000 0100 (04 hex)

Если подчиненный выполнил затребованное действие без ошибки, он возвращает такой же код. Если имеет место ошибка, то он возвращает:

1000 0100 (84 hex)

В добавление к изменению кода функции, преобразователь СН3020 размещает в поле данных уникальный код, который говорит Серверу, какая именно ошибка произошла или причину ошибки.

#### Содержание поля данных

Поле данных в сообщении от Сервера к преобразователю содержит дополнительную информацию, которая необходима преобразователю для выполнения указанной функции. Оно может содержать адреса регистров, их количество, счетчик передаваемых байтов данных.

Например, если Сервер запрашивает у преобразователя СН3020 прочитать группу регистров (код функции 04), поле данных содержит адрес начального регистра и количество регистров. Если Сервер хочет записать группу регистров (код функции 10 hex), поле данных содержит адрес начального регистра, количество регистров, счетчик количества байтов данных и данные для записи в регистры.

Поле данных может отсутствовать (иметь нулевую длину) в определенных типах сообщений.

#### Содержание поля контрольной суммы

В MODBUS - сетях используются два метода контроля ошибок передачи. Содержание поля контрольной суммы зависит от выбранного способа передачи:

ASCII

Когда используется ASCII-режим, поле контрольной суммы содержит два ASCII-символа. Контрольная сумма является результатом вычисления LRC (Longitudinal Redundancy Check), сделанного над содержанием сообщения, исключая ":" и CR-LF.

RTU

Когда используется RTU-режим, поле контрольной суммы содержит 16-ти битовую величину. Контрольная сумма является результатом вычисления CRC (Cyclical Redundancy Check), сделанного над содержанием сообщения.

#### Формат передачи символов

Передача символов идет младшим битом вперед.

#### ASCII фрейм

#### С контролем четности

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Старт | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Паритет | Стоп |

#### Без контроля четности

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Старт | 1 | 2 | 3 | 4 | 5 | 6 | 7 | Стоп | Стоп |

#### RTU фрейм

#### С контролем четности

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Старт | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Паритет | Стоп |

#### Без контроля четности

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Старт | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Стоп | Стоп |

#### Методы контроля ошибок

Стандартная MODBUS сеть использует два метода контроля ошибок: контроль паритета (even/odd) и контрольная сумма. Обе эти проверки генерируются в Сервере. Преобразователь СН3020 проверяет каждый байт и все сообщение в процессе приема.

Если преобразователь СН3020 обнаружил ошибку передачи, то он не формирует ответ Серверу.

#### Контроль паритета

Пользователь может конфигурировать преобразователи на проверку четного или нечетного паритета (Even/Odd).

Например, 8 бит-режима содержат следующую информацию: 1100 0101. Общее количество единиц - 4. Если используется четный паритет, то бит паритета будет равен 0, и общее количество единиц будет по прежнему четным числом. Если используется нечетный паритет, то бит паритета будет равен 1, тогда общее количество единиц вместе с битом паритета будет равно пяти, т.е. нечетному числу.

#### Контрольная сумма LRC

Метод LRC проверяет содержание сообщения исключая начальный символ ":" и пару CR-LF.

LRC - один байт. LRC вычисляется передающим устройством и добавляется в конец сообщения. Принимающее устройство вычисляет LRC в процессе приема сообщения и сравнивает его с полем LRC принятого сообщения. Если есть несовпадение, то имеет место ошибка.

#### Контрольная сумма CRC

Контрольная сумма CRC состоит из двух байт. Контрольная сумма вычисляется передающим устройством и добавляется в конец сообщения. Принимающее устройство вычисляет контрольную сумму в процессе приема и сравнивает ее с полем CRC принятого сообщения.

Счетчик контрольной суммы предварительно инициализируется числом FF hex. Только восемь бит данных используются для вычисления контрольной суммы CRC. Старт и стоп биты, бит паритета, если он используется, не учитываются в контрольной сумме.

Во время генерации CRC каждый байт сообщения складывается по исключающему ИЛИ с текущим содержимым регистра контрольной суммы. Результат сдвигается в направлении младшего бита, с заполнением старшего бита нулем. Если младший бит равен 1, то производится исключающее ИЛИ содержимого регистра контрольной суммы и определенного числа. Если младший бит равен 0, то исключающее ИЛИ не делается.

Процесс сдвига повторяется восемь раз. После последнего (восьмого) сдвига, следующий байт складывается с текущей величиной регистра контрольной суммы, и процесс сдвига повторяется восемь раз как описано выше. Конечное содержание регистра и есть контрольная сумма CRC.

### Номенклатура выходных параметров

Преобразователь СН3020 обеспечивает выдачу значений параметров измеряемых величин. Номенклатура измеряемых параметров определяется исполнением преобразователя СН3020. Модификации преобразователей СН3020 и перечень измеряемых параметров приведены в таблице Г.4.

Таблица Г.4

| Наименование параметра | Обo-значение | Измеряемые параметры | | | |
| --- | --- | --- | --- | --- | --- |
| СН3020/1-4-ХХХ-Х | СН3020/1-3-ХХХ-Х | СН3020/2-4-ХХХ | СН3020/2-3-ХХХ |
| Суммарная активная мощность | P | P | P | - | - |
| Активная мощность фазы нагрузки | Pa  Pb  Pc | Pa  Pb  Pc | -  -  - | -  -  - | -  -  - |
| Суммарная реактивная мощность | Q | Q | Q | - | - |
| Реактивная мощность фазы нагрузки | Qa  Qb  Qc | Qa  Qb  Qc | -  -  - | -  -  - | -  -  - |
| Действующее значение фазного напряжения | Ua  Ub  Uc | Ua  Ub  Uc | Uab  -  Ucb | Ua  Ub  Uc | Uab  -  Ucb |
| Действующее значение междуфазного напряжения | Uab  Ubc  Uca | Uab  Ubc  Uca | -  -  - | Uab  Ubc  Uca | -  -  - |
| Действующее значение фазного тока | Ia  Ib  Ic | Ia  Ib  Ic | Ia  -  Ic | -  -  - | -  -  - |
| Частота сети | F | F | F | F | F |
| Суммарная полная мощность | S | S | S | - | - |
| Полная мощность фазы нагрузки | Sa  Sb  Sc | Sa  Sb  Sc | -  -  - | -  -  - | -  -  - |
| Среднее значение тока по фазам | Iср | Iср | Iср | - | - |
| Среднее значение линейного напряжения | Uлср | Uлср | Uлср | Uлср | Uлср |
| Коэффициент мощности | Kp | Kp | Kp | - | - |

Для отсутствующего параметра преобразователь СН3020 выдает значение ±∞ (бесконечность).

При необходимости, пользователь может переупорядочить параметры в блоке выходных регистров. Фактическое расположение параметра в области текущих значений выходных регистров определяется конфигурацией преобразователя СН3020.

### Формат float представления выходных значений параметров

Выходные значения измеряемых величин, коэффициенты трансформации и эталоны для калибровки каналов представляются в формате c плавающей запятой одинарной точности (float) стандарта IEEE 754 (Рисунок Г1).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Бит | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 31 |  |  |  |  |  |  | 24 | 23 |  |  |  |  |  |  | 16 | 15 |  |  |  |  |  |  | 8 | 7 |  |  |  |  |  |  | 0 |
| S | E | E | E | E | E | E | E | E | M | M | M | M | M | M | M | M | M | M | M | M | M | M | M | M | M | M | M | M | M | M | M |
|  | Экспонента | | | | | | | | Мантисса | | | | | | | | | | | | | | | | | | | | | | |
| Знак числа | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

Рисунок Г1 - Представление числа в формате c плавающей запятой одинарной

точности (float) стандарта IEEE 754

Представление числа в формате c плавающей запятой одинарной точности.

Число:

= (-1)s2e × 1.f (нормировано) если E > 0 , иначе

= (-1)s2-126 × 0.f (не нормировано),

где

f = (b23-1+b22-2+ bin +…+b0-23) где bin =1 или 0;

s = знак (0 - положительный; 1 - отрицательный);

E = смещенная экспонента;

Emax=255, Emin=0; E=255 и E=0 используются для представления специальных значений;

e - несмещенная экспонента; e = E – 127(смещение).

Данные одного канала передаются в сообщении в поле двух последовательных регистров.

### Фиксация (срез) значений

Преобразователь СН3020 обеспечивает фиксацию (срез) текущих значений в памяти по команде Сервера. Команда фиксации осуществляется записью в регистр 0 значения метки – любого значения от 0 до 65535. Метка может быть считана одновременно с фиксированными значениями и обеспечивает идентификацию среза. Метка среза может быть считана функцией Modbus 4 из регистра 100 (64 hex). Фиксированные значения располагаются в области регистров 101…144. Назначение параметров в области фиксированных значений соответствует назначению параметров в области текущих значений выходных регистров и определяется конфигурацией преобразователя СН3020.

### Чтение кода АЦП

Для чтения данных непосредственно из АЦП необходимо выбрать один из шести каналов АЦП и запустить режим циклического обновления значений (сервисная функция Modbus 8 подфункция 2). Преобразователь СН3020 обеспечивает циклическое обновление в регистре 99 (63 hex) данных выбранного канала. При этом данные выходных каналов не обновляются. Сброс режима осуществляется рестартом питания преобразователя СН3020, рестартом преобразователя СН3020 (сервисная функция Modbus 8 подфункция 1) или сбросом режима циклической выдачи (сервисная функция Modbus 8 подфункция 2 код 0).

### Рестарт интерфейсного модуля преобразователя СН3020

Рестарт интерфейсного модуля преобразователя СН3020 осуществляется рестартом питания преобразователя СН3020 или вызовом сервисной функции Modbus 8 (подфункция 1).

### Регистр состояния преобразователя СН3020

Регистр состояния преобразователя СН3020 0 отображает состояние измерительного тракта и вычислителя преобразователя СН3020. Значение регистра состояния соответствует текущему циклу измерения.

### Функции контроля и обработки данных

Сервер в отношении преобразователя СН3020 осуществляет следующие функции:

- чтение значений параметров (функция Modbus 04);

- чтение фиксированных значений (среза) параметров (функция Modbus 04);

- передача команд калибровки, фиксации данных, диагностики;

- чтение значений АЦП (в сервисном режиме);

- параметризацию преобразователя СН3020.

#### Чтение содержимого последовательности регистров

Посредством команд чтения осуществляется считывание регистров, содержащих текущие результаты измерения и фиксированные значения (срез).

В ASCII режиме за один запрос можно считывать не более 11-ти параметров (22 регистра).

#### Функция Modbus 04

Формат запроса Cервера приведен в таблице Г.5;

Таблица Г.5

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | | № байта | Значение (HEX) |
| Заголовок | Адрес преобразователя СН3020 | 0 |  |
| Функция | 1 | 04 |
| Адрес регистра, старший байт | 2 |  |
| Адрес регистра, младший байт | 3 |  |
| Число регистров, старший байт | 4 |  |
| Число регистров, младший байт | 5 |  |
| Контрольная сумма (CRC) | CRC, старший байт | 6 |  |
| CRC, младший байт | 7 |  |

Формат ответа преобразователя СН3020 приведен в таблице Г.6:

Таблица Г.6

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | | № байта | Значение (HEX) |
| Заголовок | Адрес преобразователя СН3020 | 0 |  |
| Функция | 1 | 04 |
| Число байт данных | 2 |  |
| Данные | Данные, разряды 0…7 | 3 | Float IEEE 754 |
| Данные, разряды 8…15 | 4 |
| Данные, разряды 16…23 | 5 |
| Данные, разряды 24…31 | 6 |
| … | … |  |
| Данные, разряды 0…7 | n-5 | Float IEEE 754 |
| Данные, разряды 8…15 | n-4 |
| Данные, разряды 16…23 | n-3 |
| Данные, разряды 24…31 | n-2 |
| Контрольная сумма (CRC) | CRC, старший байт | n-1 |  |
| CRC, младший байт | n |  |
| Примечание - Данные одного канала в формате float передаются в сообщении в поле двух последовательных регистров, начиная с младшего байта. | | | |

#### Запись пользовательских коэффициентов и выдача команд пользователя

Посредством команд записи осуществляется запись коэффициентов трансформации ИТ и выдача преобразователю СН3020 пользовательских команд, необходимых при эксплуатации преобразователя СН3020. При записи эталона преобразователь СН3020 автоматически производит коррекцию измерительного канала, используя значение эталона и фактически измеренное значение входного сигнала, поданного на вход соответствующего канала.

#### Функция Modbus 16 (10 Hex)

Формат запроса сервера приведен в таблице Г.7:

Таблица Г.7

| Имя поля | | № байта | Значение (HEX) |
| --- | --- | --- | --- |
| Заголовок | Адрес преобразователя СН3020 | 0 |  |
| Функция | 1 | 10 |
| Адрес регистра, старший байт | 2 |  |
| Адрес регистра, младший байт | 3 |  |
| Число регистров, старший байт | 4 | 0 |
| Число регистров, младший байт | 5 | 2 |
| Число байт данных | 6 |  |
| Данные | Данные, разряды 0…7 | 7 | Float IEEE 754 |
| Данные, разряды 8…15 | 8 |
| Данные, разряды 16…23 | 9 |
| Данные, разряды 24…31 | 10 |
| … | … |  |
| Данные, разряды 0…7 | n-5 | Float IEEE 754 |
| Данные, разряды 8…15 | n-4 |
| Данные, разряды 16…23 | n-3 |
| Данные, разряды 24…31 | n-2 |
| Контрольная сумма (CRC) | CRC, старший байт | n-1 |  |
| CRC, младший байт | n |  |

Формат ответа преобразователя СН3020 приведен в таблице Г.8:

Таблица Г.8

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | | № байта | Значение (HEX) |
| Заголовок | Адрес преобразователя СН3020 | 0 |  |
| Функция | 1 | 10 |
| Адрес регистра, старший байт | 2 |  |
| Адрес регистра, младший байт | 3 |  |
| Число регистров, старший байт | 4 | 0 |
| Число регистров, младший байт | 5 | 2 |
| Контрольная сумма (CRC) | КС, старший байт | 6 |  |
| КС, младший байт | 7 |  |

#### Сервисные команды

Сервисные команды служат для диагностики и калибровки измерительных каналов на стадии изготовления и регулировки преобразователя СН3020.

#### Функция Modbus 08

Формат запроса от Сервера к преобразователю СН3020 приведен в таблице Г.9:

Таблица Г.9

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | | № байта | Значение (HEX) |
| Заголовок | Адрес преобразователя СН3020 | 0 |  |
| Функция | 1 | 08 |
| Подфункция, старший байт | 2 |  |
| Подфункция, младший байт | 3 |  |
| Данные | Данные, старший байт | 4 |  |
| Данные, младший байт | 5 |  |
| Контрольная сумма (CRC) | CRC, старший байт | 6 |  |
| CRC, младший байт | 7 |  |

Формат ответа от преобразователя СН3020 к Серверу приведен в таблице Г.10:

Таблица Г.10

| Имя поля | | № байта | Значение (HEX) |
| --- | --- | --- | --- |
| Заголовок | Адрес преобразователя СН3020 | 0 |  |
| Функция | 1 | 08 |
| Подфункция, старший байт | 2 |  |
| Подфункция, младший байт | 3 |  |
| Данные | Данные, старший байт | 4 |  |
| Данные, младший байт | 5 |  |
| Контрольная сумма (CRC) | CRC, старший байт | 6 |  |
| CRC, младший байт | 7 |  |

#### Подфункции диагностики приведены в таблице Г.11

#### Таблица Г.11

|  |  |  |
| --- | --- | --- |
| Подфункции (функции Modbus 08hex) | | |
| Номер подфункции | Разряды | Назначение |
| 1 | Разряды 15…0 | 0 – сброс процессора преобразователя СН3020 |
| 2 | Разряды 15…0 | Циклическая выборка значений АЦП:  8000h – по каналу Ua;  8001h – по каналу Ub;  8002h – по каналу Uc;  8003h – по каналу Ia;  8004h – по каналу Ib;  8005h – по каналу Ic;  8006h – по каналу Uref;  0000h – отключение режима циклической выборки |
| Примечания:   1. Включение режима «Циклическая выборка значений АЦП» производится с ненулевым значением старшего бита старшего байта данных в кадре запроса. 2. Отключение режима производится с нулевым значением в старшем бите старшего байта данных в кадре запроса. | | |

Назначения регистров приведены в таблице Г.12:

Таблица Г.12

|  |  |  |  |
| --- | --- | --- | --- |
| Адрес регистра (hex) | Разряды | | Назначение |
| Служебный регистр. Доступен только для чтения - функция Modbus 04 | | | |
| 0000h,  00C8h | Разряд 15 | | =0 – норма;  =1 – данные результата измерения недостоверны |
| Разряд 14 | | =0 – резерв |
| Разряд 13 | | =0 – резерв |
| Разряд 12 | | =0 – резерв |
| Разряд 11 | | =0 – норма;  =1 – сбой генератора |
| Разряд 10 | | =0 – норма;  =1 – сбой EEPROM |
| Разряд 9 | | =0 – норма;  =1 – сбой синхронизации АЦП |
| Разряд 8 | | =0 – норма;  =1 – сбой программы |
| Разряд 7 | | =0 – норма;  =1 – переполнение по частоте F |
| Разряд 6 | | =0 – норма;  =1 – сбой источника опорного напряжения Uref |
| Разряд 5 | | =0 – норма;  =1 – перегрузка по напряжению, канал Uc |
| Разряд 4 | | =0 – норма;  =1 – перегрузка по напряжению, канал Ub |
|  | | Разряд 3 | =0 – норма;  =1 – перегрузка по напряжению, канал Ua |
| Разряд 2 | =0 – норма;  =1 – перегрузка по току, канал Ic |
| Разряд 1 | =0 – норма;  =1 – перегрузка по току, канал Ib |
| Разряд 0 | =0 – норма;  =1 – перегрузка по току, канал Ia |
| Выходные регистры. Доступны только для чтения - функция Modbus 04 | | | |
| 0001h,  00C9h | | Разряды 15…0 | Идентификатор преобразователя СН3020 |
| Разряды 15…8 | Символ «M» (прописная буква) в латинском регистре |
| Разряды 7…4 | Исполнение аппаратной части преобразователя:  1 - СН3020/1-4;  2 - СН3020/1-3;  3 - СН3020/2-4;  4 - СН3020/2-3 |
| Разряды 3…0 | Исполнение программной части преобразователя СН3020:  =1 |
| 0002…  0037h | |  | Область текущих значений измеряемых параметров и коэффициентов трансформации.  Назначение параметров определяется конфигурацией. Формат значений - Float IEEE 754 |

*Продолжение таблицы Г.12*

| Адрес  регистра (hex) | Разряды | Назначение |
| --- | --- | --- |
| 0002h | Разряды 31…0 | Параметр 1 |
| 0004h | Разряды 31…0 | Параметр 2 |
| 0006h | Разряды 31…0 | Параметр 3 |
| 0008h | Разряды 31…0 | Параметр 4 |
| 000Ah | Разряды 31…0 | Параметр 5 |
| 000Ch | Разряды 31…0 | Параметр 6 |
| 000Eh | Разряды 31…0 | Параметр 7 |
| 0010h | Разряды 31…0 | Параметр 8 |
| 0012h | Разряды 31…0 | Параметр 9 |
| 0014h | Разряды 31…0 | Параметр 10 |
| 0016h | Разряды 31…0 | Параметр 11 |
| 0018h | Разряды 31…0 | Параметр 12 |
| 001Ah | Разряды 31…0 | Параметр 13 |
| 001Ch | Разряды 31…0 | Параметр 14 |
| 001Eh | Разряды 31…0 | Параметр 15 |
| 0020h | Разряды 31…0 | Параметр 16 |
| 0022h | Разряды 31…0 | Параметр 17 |
| 0024h | Разряды 31…0 | Параметр 18 |
| 0026h | Разряды 31…0 | Параметр 19 |
| 0028h | Разряды 31…0 | Параметр 20 |
| 002Ah | Разряды 31…0 | Параметр 21 |
| 002Ch | Разряды 31…0 | Параметр 22 |
| 002Eh | Разряды 31…0 | Параметр 23 |
| 0030h | Разряды 31…0 | Параметр 24 |
| 0032h | Разряды 31…0 | Параметр 25 |
| 0034h | Разряды 31…0 | Параметр 26 |
| 0036h | Разряды 31…0 | Параметр 27 |
|  |  | Резерв |
| 0063h | Разряды 15…0 | А - данные с выхода АЦП по выбранному каналу (запуск режима циклической выборки значений и выбор канала осуществляются через сервисную функцию) |
| 0064h | Разряды 15…0 | Метка среза, установленная Сервером |
| 0065…  …009Ah |  | Область фиксированных значений измеряемых параметров (данные среза).  Назначение параметров определяется конфигурацией. Формат значений - Float IEEE 754 |
| 0065h | Разряды 31…0 | Параметр 1 |
| 0067h | Разряды 31…0 | Параметр 2 |
| 0069h | Разряды 31…0 | Параметр 3 |
| 006Bh | Разряды 31…0 | Параметр 4 |
| 006Dh | Разряды 31…0 | Параметр 5 |
| 006Fh | Разряды 31…0 | Параметр 6 |
| 0071h | Разряды 31…0 | Параметр 7 |
| 0073h | Разряды 31…0 | Параметр 8 |
| 0075h | Разряды 31…0 | Параметр 9 |
|  |  |  |

*Продолжение таблицы Г.12*

| Адрес  регистра (hex) | Разряды | Назначение | | | |
| --- | --- | --- | --- | --- | --- |
| 0077h | Разряды 31…0 | Параметр 10 | | | |
| 0079h | Разряды 31…0 | Параметр 11 | | | |
| 007Bh | Разряды 31…0 | Параметр 12 | | | |
| 007Dh | Разряды 31…0 | Параметр 13 | | | |
| 007Fh | Разряды 31…0 | Параметр 14 | | | |
| 0081h | Разряды 31…0 | Параметр 15 | | | |
| 0083h | Разряды 31…0 | Параметр 16 | | | |
| 0085h | Разряды 31…0 | Параметр 17 | | | |
| 0087h | Разряды 31…0 | Параметр 18 | | | |
| 0089h | Разряды 31…0 | Параметр 19 | | | |
| 008Bh | Разряды 31…0 | Параметр 20 | | | |
| 008Dh | Разряды 31…0 | Параметр 21 | | | |
| 008Fh | Разряды 31…0 | Параметр 22 | | | |
| 0091h | Разряды 31…0 | Параметр 23 | | | |
| 0093h | Разряды 31…0 | Параметр 24 | | | |
| 0095h | Разряды 31…0 | Параметр 25 | | | |
| 0097h | Разряды 31…0 | Параметр 26 | | | |
| 0099h | Разряды 31…0 | Параметр 27 | | | |
|  |  |  | | | |
| 00C8…  00FFh | Область фиксированных значений измеряемых параметров.  Float IEEE 754 | | | | |
|  |  | СН3020/1-4 | СН3020/1-3 | СН3020/2-4 | СН3020/2-3 |
| 00CAh | Разряды 31…0 | P | P | - | - |
| 00CCh | Разряды 31…0 | Pa | - | - | - |
| 00CEh | Разряды 31…0 | Pb | - | - | - |
| 00D0h | Разряды 31…0 | Pc | - | - | - |
| 00D2h | Разряды 31…0 | Q\_ | Q | - | - |
| 00D4h | Разряды 31…0 | Qa | - | - | - |
| 00D6h | Разряды 31…0 | Qb | - | - | - |
| 00D8h | Разряды 31…0 | Qc | - | - | - |
| 00DAh | Разряды 31…0 | Ua | Uab | Ua | Uab |
| 00DCh | Разряды 31…0 | Ub | - | Ub | - |
| 00DEh | Разряды 31…0 | Uc | Ucb | Uc | Ucb |
| 00E0h | Разряды 31…0 | Uaв | - | Uaв | - |
| 00E2h | Разряды 31…0 | Uac | - | Uac | - |
| 00E4h | Разряды 31…0 | Ubc | - | Ubc | - |
| 00E6h | Разряды 31…0 | Ia | Ia | - | - |
| 00E8h | Разряды 31…0 | Ib | - | - | - |
| 00EAh | Разряды 31…0 | Ic | Ic | - | - |
| 00ECh | Разряды 31…0 | F | F | F | F |
| 00EEh | Разряды 31…0 | S\_ | S | - | - |
| 00F0h | Разряды 31…0 | Sa | - | - | - |
| 00F2h | Разряды 31…0 | Sb | - | - | - |
| 00F4h | Разряды 31…0 | Sc | - | - | - |
| 00F6h | Разряды 31…0 | Кн | Кн | Кн | Кн |
| 00F8h | Разряды 31…0 | Кт | Кт | Кт | Кт |

*Окончание таблицы Г.12*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Адрес  регистра (hex) | Разряды | | Назначение | | | | |
| 00FAh | | Разряды 31…0 | | Iср | Iср | - | - |
| 00FCh | | Разряды 31…0 | | Uлср | Uлср | Uлср | Uлср |
| 00FEh | | Разряды 31…0 | | Kp | Kp | - | - |
|  | | | | | | | |
|  | | | | | | | |
| Регистры эталонов. Доступны для чтения – функция Modbus 3 (03hex) | | | | | | | |
| 0004h | Разряды 31…0 | | Кн – коэффициент трансформации по напряжению, формат float | | | | |
| 0006h | Разряды 31…0 | | Кт – коэффициент трансформации по току, формат float | | | | |
| 0016h | Разряды 31…0 | | Кp – коэффициент мощности, формат float | | | | |
| 0020…  …003Fh |  | | Строка пользовательских данных (32 байта) | | | | |
|  | | | | | | | |
| Регистры эталонов. Доступны для записи – функция Modbus 16 (10hex) | | | | | | | |
| 0000h | Разряды 15…0 | | =0…65535 – метка среза.  При записи метки осуществляется фиксация текущих значений (срез) | | | | |
| 0002h | Разряды 31…0 | | Резерв | | | | |
| 0004h | Разряды 31…0 | | Кн – коэффициент трансформации по напряжению, формат float | | | | |
| 0006h | Разряды 31…0 | | Кт – коэффициент трансформации по току, формат float | | | | |
| 0008h | Разряды 31…0 | | Fet – эталон частоты, формат float | | | | |
| 000Ah | Разряды 31…0 | | Uaet – эталон напряжения для канала Ua, формат float | | | | |
| 000Ch | Разряды 31…0 | | Ubet – эталон напряжения для канала Ub, формат float | | | | |
| 000Eh | Разряды 31…0 | | Ucet – эталон напряжения для канала Uc, формат float | | | | |
| 0010h | Разряды 31…0 | | Iaet – эталон тока для канала Ia, формат float | | | | |
| 0012h | Разряды 31…0 | | Ibet – эталон тока для канала Ib, формат float | | | | |
| 0014h | Разряды 31…0 | | Icet – эталон тока для канала Ic, формат float | | | | |

**Функция выдачи на табло**

Преобразователь СН3020 имеет возможность циклически выдавать через любой из стыков значения измеренных параметров для отображения на внешних цифровых табло.

Для того чтобы преобразователь СН3020 выдавал на табло необходимо настроить параметры порта:

- протокол – ModBus,

- способ передачи – циклическая передача,

- скорость – скорость передачи данных, установленная на табло.

Для выдачи данных на табло используется функция Modbus 16 (10 Hex).

Формат выдачи данных от преобразователя СН3020 к табло приведен в таблице Г.13:

Таблица Г.13

| Имя поля | | | | | №  байта | Значение (HEX) |
| --- | --- | --- | --- | --- | --- | --- |
| Заголовок | Адрес табло | | | | 0 | 0 |
| Функция | | | | 1 | 10 |
| Адрес регистра, старший байт | | | | 2 | 0 |
| Адрес регистра, младший байт | | | | 3 | 0 |
| Число регистров, старший байт | | | | 4 | 0 |
| Число регистров, младший байт | | | | 5 | 2C |
| Число байт данных | | | | 6 | 58 |
| Данные | Служебный регистр | | | | 7,8 |  |
| Идентификатор преобразователя СН3020 (см. регистр 0001) | | | | 9,10 |  |
| СН3020/1-4 | СН3020/1-3 | СН3020/2-4 | СН3020/2-3 |  |  |
| P | P | - | - | 11…14 | Float IEEE 754 |
| Pa | - | - | - | 15…18 | Float IEEE 754 |
| Pc | - | - | - | 23…26 | Float IEEE 754 |
| Q | Q | - | - | 27…30 | Float IEEE 754 |
| Qa | - | - | - | 31…34 | Float IEEE 754 |
| Qb | - | - | - | 35…38 | Float IEEE 754 |
| Qc | - | - | - | 39…42 | Float IEEE 754 |
| Ua | Uab | Ua | Uab | 43…46 | Float IEEE 754 |
| Ub | - | Ub | - | 47…50 | Float IEEE 754 |
| Uc | Ucb | Uc | Ucb | 51…54 | Float IEEE 754 |
| Uab | - | Uaв | - | 55…58 | Float IEEE 754 |
| Uac | - | Uac | - | 59…62 | Float IEEE 754 |
| Ubc | - | Ubc | - | 63…66 | Float IEEE 754 |
| Ia | Ia | - | - | 67…70 | Float IEEE 754 |
| Ib | - | - | - | 71…74 | Float IEEE 754 |
| Ic | Ic | - | - | 75…78 | Float IEEE 754 |
| F | F | F | F | 79…82 | Float IEEE 754 |
| S | S | - | - | 83…86 | Float IEEE 754 |
| Sa | - | - | - | 87…90 | Float IEEE 754 |
| Sb | - | - | - | 91…94 | Float IEEE 754 |
| Sc | - | - | - | 95…98 | Float IEEE 754 |
| Контрольная сумма (CRC) | CRC, старший байт | | | | 99 |  |
| CRC, младший байт | | | | 100 |  |

Параметризация преобразователя СН3020.

Параметризация преобразователя СН3020 возможна только в протоколе ModBus RTU.

После подачи питания на преобразователь СН3020, в течении первых 5-ти секунд по обоим портам устанавливается режим работы с конфигуратором:

- способ передачи – по запросу;

- скорость – 57600 б/с;

- адрес – 1;

- протокол – ModBus RTU;

- четность – нет;

При получении любой корректной посылки, т.е. совпадает контрольная сумма и адрес, время режима работы с конфигуратором продлевается на 10 секунд. При каждом очередном корректном запросе время продлевается еще на 10 секунд. Если в течении этого времени не было ни одного запроса, то преобразователь переходит в рабочий режим и активизирует параметры, установленные конфигуратором.

Функция записи параметров доступна только в режиме работы с конфигуратором.

Для параметризации интерфейсного узла преобразователя СН3020 используется две функции: функция ModBus 20 - используется для чтения параметров, и функция ModBus 21 - используется для записи параметров.

#### Функция Modbus 20 (14 Hex)

Формат запроса сервера к преобразователю СН3020 приведен в таблице Г.14:

Таблица Г.14

| Имя поля | | № байта | Значение (HEX) |
| --- | --- | --- | --- |
| Заголовок | Адрес преобразователя СН3020 | 0 |  |
| Функция | 1 | 14 |
| Число байт данных | 2 | 7 |
| Резерв | 3 | 0 |
| Резерв | 4 | 0 |
| Резерв | 5 | 0 |
| Адрес, старший байт | 6 |  |
| Адрес, младший байт | 7 |  |
| Число слов, старший байт | 8 |  |
| Число слов, младший байт | 9 |  |
| Контрольная сумма (CRC) | CRC, старший байт | 10 |  |
| CRC, младший байт | 11 |  |

Формат ответа преобразователя СН3020 Серверу приведен в таблице Г.15:

Таблица Г.15

|  |  |  |  |
| --- | --- | --- | --- |
| Имя поля | | № байта | Значение (HEX) |
| Заголовок | Адрес преобразователя СН3020 | 0 |  |
| Функция | 1 | 14 |
| Число байт данных + 1 | 2 |  |
| Число байт данных | 3 |  |
| Данные | Резерв | 4 |  |
|  | Данные | 5 |  |
| … | … |  |
| Данные | n-2 |  |
| Контрольная сумма (CRC) | CRC, старший байт | n-1 |  |
| CRC, младший байт | n |  |

#### Функция Modbus 21 (15 Hex)

Формат запроса Сервера к преобразователю СН3020 приведен в таблице Г.16:

Таблица Г.16

| Имя поля | | № байта | Значение (HEX) |
| --- | --- | --- | --- |
| Заголовок | Адрес преобразователя СН3020 | 0 |  |
| Функция | 1 | 15 |
| Число байт данных | 2 | 7 |
| Резерв | 3 | 0 |
| Резерв | 4 | 0 |
| Резерв | 5 | 0 |
| Адрес, старший байт | 6 |  |
| Адрес, младший байт | 7 |  |
| Число слов, старший байт | 8 |  |
| Число слов, младший байт | 9 |  |
| Данные | Данные | 10 |  |
| … | … |  |
| Данные | n-2 |  |
| Контрольная сумма (CRC) | CRC, старший байт | n-1 |  |
| CRC, младший байт | n |  |

От преобразователя СН3020 к Серверу:

Ответ повторяет запрос сервера.

В таблице Г.17 приведены параметры интерфейсного модуля

Таблица Г.17

| Адрес регистра (hex) | Разряды | Назначение (значение) |
| --- | --- | --- |
| 0000h | Разряды 7…0 | Адрес преобразователя СН3020 на порту COM1: от 0 до 255 |
| 0001h | Разряды 7…0 | Адрес преобразователя СН3020 на порту COM2: от 0 до 255 |
| 0002h | Разряды 7…0 | =1 – служебный параметр |
| 0003h | Разряды 7…0 | Скорость передачи на порту COM1:  =0 - 110 бит/с;  =1 - 150 бит/с;  =2 - 300 бит/с;  =3 - 600 бит/с;  =4 - 1200 бит/с;  =5 - 2400 бит/с;  =6 - 4800 бит/с;  =7 – 9600 бит/с;  =8 – 19200 бит/с;  =9 – 38400 бит/с;  =10 – 57600 бит/с;  =11 – 115200 бит/с; |
| 0004h | Разряды 7…0 | Скорость передачи на порту COM2:  =0 – 110 бит/с;  =1 – 150 бит/с;  =2 – 300 бит/с;  =3 – 600 бит/с;  =4 – 1200 бит/с;  =5 – 2400 бит/с;  =6 – 4800 бит/с;  =7 – 9600 бит/с;  =8 – 19200 бит/с;  =9 – 38400 бит/с;  =10 – 57600 бит/с;  =11 – 115200 бит/с; |
| 0005h | Разряды 7…0 | =15 – служебный регистр |
| 0006h | Разряды 7…0 | Способ передачи на порту COM1:  =0 – Циклическая  =6 – По запросу |
| 0007h | Разряды 7…0 | Способ передачи на порту COM2:  =0 – Циклическая  =6 – По запросу |
| 0008h | Разряды 7…0 | =0 – служебный параметр |
| 0009h | Разряды 7…0 | Протокол на порту COM1:  =2 – ModBus  =3 – ModBus ASCII  =5 – FT1.2 3020 |

*Продолжение таблицы Г.17*

| Адрес регистра (hex) | Разряды | Назначение (значение) |
| --- | --- | --- |
| 000Ah | Разряды 7…0 | Протокол на порту COM2:  =2 – ModBus  =3 – ModBus ASCII  =5 – FT1.2 3020 |
| 000Bh | Разряды 7…0 | =2 – служебный параметр |
| 000Ch | Разряды 7…0 | Формат кадра на порту COM1 |
|  | Разряды 7…0 | Четность:  =0 – Нет;  =1 – Четность;  =2 – Нечетность |
| 000Dh | Разряды 7…0 | Формат кадра на порту COM2 |
|  | Разряды 7…0 | Четность:  =0 – Нет;  =1 – Четность;  =2 – Нечетность |
| 000Eh | Разряды 7…0 | =0 – служебный параметр |
| 000Fh | Разряды 15…0 | Исполнение устройства (запись – любое значение,  чтение – значение, приведенное ниже) |
| Разряды 15…8 | Символ «M» (прописная буква) в латинском регистре |
| Разряды 7…4 | Исполнение аппаратной части преобразователя СН3020:  1 - СН3020/1-4;  2 - СН3020/1-3;  3 - СН3020/2-4;  4 - СН3020/2-3 |
| Разряды 3…0 | Исполнение программной части преобразователя СН3020:  =0 |
| 0011h | Разряды 7…0 | Индекс выходного параметра регистра 1 (Таблица Г.17а) |
| 0012h | Разряды 7…0 | Индекс выходного параметра регистра 2 (Таблица Г.17а) |
| 0013h | Разряды 7…0 | Индекс выходного параметра регистра 3 (Таблица Г.17а) |
| 0014h | Разряды 7…0 | Индекс выходного параметра регистра 4 (Таблица Г.17а) |
| 0015h | Разряды 7…0 | Индекс выходного параметра регистра 5 (Таблица Г.17а) |
| 0016h | Разряды 7…0 | Индекс выходного параметра регистра 6 (Таблица Г.17а) |
| 0017h | Разряды 7…0 | Индекс выходного параметра регистра 7 (Таблица Г.17а) |
| 0018h | Разряды 7…0 | Индекс выходного параметра регистра 8 (Таблица Г.17а) |
| 0019h | Разряды 7…0 | Индекс выходного параметра регистра 9 (Таблица Г.17а) |
| 001Ah | Разряды 7…0 | Индекс выходного параметра регистра 10 (Таблица Г.17а) |
| 001Bh | Разряды 7…0 | Индекс выходного параметра регистра 11 (Таблица Г.17а) |

*Окончание таблицы Г.17*

| Адрес регистра (hex) | Разряды | Назначение (значение) |
| --- | --- | --- |
| 001Ch | Разряды 7…0 | Индекс выходного параметра регистра 12 (Таблица Г.17а) |
| 001Dh | Разряды 7…0 | Индекс выходного параметра регистра 13 (Таблица Г.17а) |
| 001Eh | Разряды 7…0 | Индекс выходного параметра регистра 14 (Таблица Г.17а) |
| 001Fh | Разряды 7…0 | Индекс выходного параметра регистра 15 (Таблица Г.17а) |
| 0020h | Разряды 7…0 | Индекс выходного параметра регистра 16 (Таблица Г.17а) |
| 0021h | Разряды 7…0 | Индекс выходного параметра регистра 17 (Таблица Г.17а) |
| 0022h | Разряды 7…0 | Индекс выходного параметра регистра 18 (Таблица Г.17а) |
| 0023h | Разряды 7…0 | Индекс выходного параметра регистра 19 (Таблица Г.17а) |
| 0024h | Разряды 7…0 | Индекс выходного параметра регистра 20 (Таблица Г.17а) |
| 0025h | Разряды 7…0 | Индекс выходного параметра регистра 21 (Таблица Г.17а) |
| 0026h | Разряды 7…0 | Индекс выходного параметра регистра 22 (Таблица Г.17а) |

Таблица Г.17а

| Индекс выходного параметра, разряды 7..0 | Возвращаемые значения | | | |
| --- | --- | --- | --- | --- |
| СН3020/1-4 | СН3020/1-3 | СН3020/2-4 | СН3020/2-3 |
| 0 | P | P | - | - |
| 1 | Pa | - | - | - |
| 2 | Pb | - | - | - |
| 3 | Pc | - | - | - |
| 4 | Q | Q | - | - |
| 5 | Qa | - | - | - |
| 6 | Qb | - | - | - |
| 7 | Qc | - | - | - |
| 8 | Ua | Uab | Ua | Uab |
| 9 | Ub | - | Ub | - |
| 10 | Uc | Ubc | Uc | Ubc |

*Окончание таблицы Г.17а*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Индекс выходного параметра, разряды 7..0 | Возвращаемые значения | | | |
| СН3020/1-4 | СН3020/1-3 | СН3020/2-4 | СН3020/2-3 |
| 11 | Uab | - | Uab | - |
| 12 | Uac | - | Uac | - |
| 13 | Ucb | - | Ucb | - |
| 14 | Ia | Ia | - | - |
| 15 | Ib | - | - | - |
| 16 | Ic | Ic | - | - |
| 17 | F | F | F | F |
| 18 | S | S | - | - |
| 19 | Sa | - | - | - |
| 20 | Sb | - | - | - |
| 21 | Sc | - | - | - |
| 22 | Iср | Iср | - | - |
| 23 | Uлср | Uлср | Uлср | Uлср |
| 24 | Kp | Kp | - | - |

Информационный обмен в протоколе FT1.2

Информационный обмен между сервером и преобразователем СН3020 осуществляется по средствам кадров (посылок) постоянной длины формата FT 1.2 (ГОСТ Р МЭК 870-5-2). Размер отдельного кадра определяется только направлением передачи информации:

– при передаче информации от сервера к преобразователю кадр имеет размер 8 байт;

– при передаче информации от преобразователя СН3020 к серверу – 10 байт.

Под информацией понимаются как результаты измерений, отсылаемые преобразователем СН3020 серверу, так и управляющие команды, и запросы сервера к преобразователю. Используемый протокол обмена – последовательный, по стандарту RS485 со следующими характеристиками:

– скорость обмена 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 бит/с (устанавливается программно при параметризации преобразователя СН3020);

– число бит данных – 8;

– контроль четности отсутствует;

– число стоп-бит – 1;

Общий формат кадров сервера представлен в таблице Г.18.

Таблица Г.18

|  |  |  |
| --- | --- | --- |
| Номер  байта | Значение | Комментарий |
| 1 | 10h | Старт- байт |
| 2 | Address | Адрес преобразователя СН3020 |
| 3 | Function | Код функции; |
| 4 | Mant.Low | Младший байт мантиссы данных |
| 5 | Mant.High | Старший байт мантиссы данных |
| 6 | EXP | Экспонента мантиссы данных |
| 7 | CRC | Контрольная сумма байтов 2,3 ... 6 по модулю 256 |
| 8 | 16h | Стоп-байт |

Достоверность обращения сервера контролируется преобразователем СН3020 путём сравнения поступающего кадра с маской, содержащей: старт-байт, адрес, контрольную сумму, стоп-байт. При несовпадении кадра с маской на некотором байте, преобразователь СН3020 настраивается на прием нового кадра.

В зависимости от кода функции Function преобразователь СН3020 может игнорировать содержимое полей Mant.Low, Mant.High и EXP в кадре сервера. Значения в формате с плавающей запятой определяются соотношением:



(Г.1)



где – мантисса числа – знаковое целое 16-ти разрядное;



– экспонента числа – знаковое целое 8-ми разрядное.

Если код функции двух байтный, то второй байт передается в поле Mant.Low, значения неиспользуемых байтов - произвольное.

Перечень кодов функций и возвращаемые значения в зависимости от модификации преобразователя СН3020 представлены в таблице Г.19.

Таблица Г.19

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование  функции | Код функции | Примечание | Возвращаемые значения | | | |
| СН3020/1-4 | СН3020/1-3 | СН3020/2-4 | СН3020/2-3 |
| «Запрос результата измерения P» | 50h, 5Fh | ASCII-код символа “P,\_” | P | P | - | - |
| «Запрос результата измерения Pa» | 50h, 61h | ASCII-код символа “P,a” | Pa | - | - | - |
| «Запрос результата измерения Pb» | 50h, 62h | ASCII-код символа “P,b” | Pb | - | - | - |
| «Запрос результата измерения Pc» | 50h, 63h | ASCII-код символа “P,c” | Pc | - | - | - |
| «Запрос результата измерения Q» | 51h, 5Fh | ASCII-код символа “Q,\_” | Q | Q | - | - |
| «Запрос результата измерения Qa» | 51h, 61h | ASCII-код символа “Q,a” | Qa | - | - | - |
| «Запрос результата измерения Qb» | 51h, 62h | ASCII-код символа “Q,b” | Qb | - | - | - |
| «Запрос результата измерения Qc» | 51h, 63h | ASCII-код символа “Q,c” | Qc | - | - | - |
| «Запрос результата измерения S» | 53h, 5Fh | ASCII-код символа “S,\_” | S | S | - | - |
| «Запрос результата измерения Qa» | 51h, 61h | ASCII-код символа “Q,a” | Qa | - | - | - |
| «Запрос результата измерения Qb» | 51h, 62h | ASCII-код символа “Q,b” | Qb | - | - | - |
| «Запрос результата измерения Qc» | 51h, 63h | ASCII-код символа “Q,c” | Qc | - | - | - |
| «Запрос результата измерения S» | 53h, 5Fh | ASCII-код символа “S,\_” | S | S | - | - |
| «Запрос результата измерения Sa» | 53h, 61h | ASCII-код символа “S,a” | Sa | - | - | - |
| «Запрос результата измерения Sb» | 53h, 62h | ASCII-код символа “S,b” | Sb | - | - | - |
| «Запрос результата измерения Sc» | 53h, 63h | ASCII-код символа “S,c” | Sc | - | - | - |
| «Запрос среднего значения линейного напряжения Uлср» | 55h, FFh | ASCII-код символа “U, ” | Uлср | Uлср | Uлср | Uлср |
| «Запрос результата измерения Ua» | 55h, 61h | ASCII-код символа “U,a” | Ua | Uab | Ua | Uab |
| «Запрос результата измерения Ub» | 55h, 62h | ASCII-код символа “U,b” | Ub | - | Ub | - |

*Продолжение таблицы Г.19*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование  функции | Код функции | Примечание | Возвращаемые значения | | | |
| СН3020/1-4 | СН3020/1-3 | СН3020/2-4 | СН3020/2-3 |
| «Запрос результата измерения Uc» | 55h, 63h | ASCII-код символа “U,c” | Uc | Ucb | Uc | Ucb |
| «Запрос результата измерения Uab» | 55h, 41h | ASCII-код символа “U,A”. | Uab | - | Uab | - |
| «Запрос результата измерения Ubc» | 55h, 42h | ASCII-код символа “U,B”. | Ubc | - | Ubc | - |
| «Запрос результата измерения Uac» | 55h, 43h | ASCII-код символа “U,C” | Uac | - | Uac | - |
| «Запрос среднего значения тока Iср» | 49h, 5Fh | ASCII-код символа “I,\_” | I | I | - | - |
| «Запрос результата измерения Ia» | 49h, 61h | ASCII-код символа “I,a” | Ia | Ia | - | - |
| «Запрос результата измерения Ib» | 49h, 62h | ASCII-код символа “I,b” | Ib | - | - | - |
| «Запрос результата измерения Ic» | 49h, 63h | ASCII-код символа “I,c” | Ic | Ic | - | - |
| «Запрос результата измерения F» | 46h | ASCII-код символа “F” | F | F | F | F |
| «Запрос коэффициента мощности Kp» | 4Bh, 5Fh | ASCII-код символа “K,\_” | Kp | Kp | - | - |
| Широковещательная команда «Сохранить результат измерения» (срез) | 77h | ASCII-код символа “w” |  |  |  |  |
| «Запрос сохраненного результата измерения (среза) P» | 70h, 5Fh | ASCII-код символа “p, \_” | P | P | - | - |
| «Запрос сохраненного результата измерения (среза) Pa» | 70h, 61h | ASCII-код символа “p, a” | Pa | - | - | - |
| «Запрос сохраненного результата измерения (среза) Pb» | 70h, 62h | ASCII-код символа “p, b” | Pb | - | - | - |
| «Запрос сохраненного результата измерения(среза) P» | 70h, 63h | ASCII-код символа “p, c” | Pc | - | - | - |
| «Запрос сохраненного результата измерения (среза) Q»” | 71h, 5Fh | ASCII-код символа “q, \_” | Q | Q | - | - |

*Продолжение таблицы Г.19*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование  функции | Код функции | Примечание | Возвращаемые значения | | | |
| СН3020/1-4 | СН3020/1-3 | СН3020/2-4 | СН3020/2-3 |
| «Запрос сохраненного результата измерения (среза) Qa» | 71h, 61h | ASCII-код символа “q, a” | Qa | - | - | - |
| «Запрос сохраненного результата измерения  (среза) Qb» | 71h, 62h | ASCII-код символа “q, b” | Qb | - | - | - |
| «Запрос сохраненного результата измерения (среза) Qc» | 71h, 63h | ASCII-код символа “q, c”. | Qc | - | - | - |
| «Запрос сохраненного результата измерения (среза) S» | 73h, 5Fh | ASCII-код символа “s,\_” | S | S | - | - |
| «Запрос сохраненного результата измерения (среза) Sa» | 73h, 61h | ASCII-код символа “s,a” | Sa | - | - | - |
| «Запрос сохраненного результата измерения (среза) Sb» | 73h, 62h | ASCII-код символа “s,b” | Sb | - | - | - |
| «Запрос сохраненного результата измерения (среза) Sc» | 73h, 63h | ASCII-код символа “s,c” | Sc | - | - | - |
| «Запрос сохраненного результата измерения (среза) U»” | 75h, 61h | ASCII-код символа “u, a” | Ua | Uab | Ua | Uab |
| «Запрос сохраненного результата измерения  (среза) U»” | 75h, 62h | ASCII-код символа “u, b” | Ub | - | Ub | - |
| «Запрос сохраненного результата измерения (среза) U»” | 75h, 63h | ASCII-код символа “u, c” | Uc | Ucb | Uc |  |
| «Запрос сохраненного результата измерения  (среза) Uab» | 75h, 41h | ASCII-код символа “u,A” | Uab | - | Uab |  |
| «Запрос сохраненного результата измерения (среза) Ubc» | 75h, 42h | ASCII-код символа “u,B” | Ubc | - | Ubc |  |
| «Запрос сохраненного результата измерения (среза) Uac» | 75h, 43h | ASCII-код символа “u,C” | Uac | - | Uac |  |

*Окончание таблицы Г.19*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Наименование  функции | Код функции | Примечание | Возвращаемые значения | | | |
| СН3020/1-4 | СН3020/1-3 | СН3020/2-4 | СН3020/2-3 |
| «Запрос сохраненного результата измерения (среза) Ia» | 69h, 61h | ASCII-код символа “i, a” | Ia | Ia | - |  |
| «Запрос сохраненного результата измерения (среза) I»” | 69h, 62h | ASCII-код символа “i, b” | Ib | - | - |  |
| «Запрос сохраненного результата измерения (среза) Ic» | 69h, 63h | ASCII-код символа “i, c” | Ic | Ic | - |  |
| «Запрос сохраненного результата измерения (среза) F» | 66h | ASCII-код символа “f” | F | F | F |  |
| «Чтение коэффициента трансформации Kн» | 91h |  | Kн | Kн | Kн |  |
| «Чтение коэффициента трансформации Kт» | 92h |  | Kт | Kт | - |  |
| «Чтение пользовательских данных» | 9Eh |  | + | + | + |  |

При эксплуатации в составе телемеханического комплекса преобразователь СН3020 работает в режиме запрос/ответ. Используется основные рабочие функции «Запрос результатов измерения".

При получении кадра с кодом функции «Запрос результата измерения» преобразователь СН3020 начинает формировать 10-ти байтовый ответ серверу сразу после завершения формирования сервером последнего байта в кадре запроса. Формат кадра ответа преобразователя СН3020 на запрос сервера приводится в таблице Г.20.

Таблица Г.20

|  |  |  |
| --- | --- | --- |
| Номер байта | Значение | Комментарий |
| 1 | 10h | Старт-байт |
| 2 | Address | Собственный адрес преобразователя СН3020 |
| 3 | Function | Повторяет код функции запроса |
| 4 | Flags.Low | Младший байт слова состояния преобразователя СН3020 |
| 5 | Flags.High | Старший байт слова состояния преобразователя СН3020 |
| 6 | Mant.Low | Младший байт мантиссы результата измерения |
| 7 | Mant.High | Старший байт мантиссы результата измерения |
| 8 | EXP | Экспонента мантиссы результата измерения |
| 9 | CRC | Контрольная сумма байтов 2,3 ... 8 по модулю 256 |
| 10 | 16h | Стоп-кадр |

Слово состояния, возвращаемое серверу в полях Flags.Low и Flags.High, имеет битовый характер и изменяется во внутреннем цикле самодиагностики преобразователя СН3020. Его формат представлен в таблице Г.21:

Таблица Г.21

|  |  |
| --- | --- |
| РазрядыFlags | Утверждение, истинность которого идентифицируется единичным состоянием соответствующего разряда Flags |
| 0 | Перегрузка по току, канал Ia |
| 1 | Перегрузка по току, канал Ib |
| 2 | Перегрузка по току, канал Ic |
| 3 | Перегрузка по напряжению, канал Ua |
| 4 | Перегрузка по напряжению, канал Ub |
| 5 | Перегрузка по напряжению, канал Uc |
| 6 | Сбой источника опорного напряжения Uref |
| 7 | Переполнение по частоте F |
| 8 | Сбой программы |
| 9 | Сбой синхронизации АЦП |
| 10 | Сбой EEPROM |
| 11 | Сбой генератора |
| 12 | - |
| 13 | - |
| 14 | - |
| 15 | Данные результата измерений не достоверны |

Результат измерения, возвращаемый серверу в полях Mant.Low, Mant.High и EXP определяется соотношением (Г.1) и выражен в основных единицах измерения – вольтах, амперах, ваттах, варах или герцах. Абсолютное значение мантиссы находится в интервале от 16384 до 32768, то есть мантисса результата измерения выровнена по старшему незнаковому разряду знакового 16-ти разрядного числа. При этом относительная погрешность представления числа результата измерения не превышает ±0.003%.

После ответа на запрос сервера преобразователя СН3020 вновь настраиваются на приём нового кадра.

Для функции чтения коэффициента трансформации его текущее значение возвращается в кадре ответа преобразователя СН3020 в формате (Г.1). Преобразователь СН3020 формирует ответ так же, как и при запросе результата измерения.

Для функции “Чтение пользовательских данных” адрес читаемой ячейки (от 0 до 31) передаётся в поле Mant.Low кадра сервера, а её содержимое возвращается в поле Mant.Low кадра ответа преобразователя СН3020. В поле Mant.High возвращается тип преобразователя СН3020, в данном случае – ASCII-код символа “M” (4Dh). В поле EXP – в старшей тетраде модификация преобразователя СН3020 (10h – СН3020/1-4, 20h – СН3020/1-3, 30h – СН3020/2-4, 40h- СН3020/2-3), в младшей тетраде – версия программы преобразователя СН3020. Преобразователь СН3020 формирует ответ так же, как и при запросе результата измерения.

Широковещательная команда «Сохранить результат измерения» (срез) с адресом равным 250, заставляет все приборы, получившие эту команду, сохранить текущие измеренные значения.

В поле Mant.Low передается идентификатор. Поле EXP игнорируется. Кадр ответа не предусмотрен.

Для функции “Запрос сохраненного результата измерения (среза)” результат измерения, возвращаемый серверу в полях Mant.Low, Mant.High и EXP определяется соотношением (1) и выражен в основных единицах измерения. В поле Flags.Low возвращается идентификатор переданный широковещательной командой «Сохранить результат измерения».

### ГЕНЕРАЦИЯ LRC/CRC

#### Генерация LRC

Longitudinal Redundancy Check(LRC) это один байт. LRC вычисляется передающим устройством и добавляется к концу сообщения. Принимающее устройство также вычисляет LRC в процессе приема и сравнивает вычисленную величину с полем контрольной суммы пришедшего сообщения. Если суммы не совпали - то имеет место ошибка.

LRC вычисляется сложением последовательности байтов сообщения, отбрасывая все переносы, и затем двойным дополнением результата. LRC - это 8-ми битовое поле, где каждое новое прибавление символа, приводящее к результату более чем 255, приводит к простому перескакиванию через 0. Так как это поле не является 9-ти битовым, перенос отбрасывается автоматически.

#### Алгоритм генерации LRC

1. Сложить все байты сообщения, исключая стартовый символ ':' и конечные CR-LF, складывая их так, чтобы перенос отбрасывался (по модулю 256).

2. Отнять получившееся значение от числа FF (Hex) - это является первым дополнением.

3. Прибавить к получившемуся значению 1 - это второе дополнение.

#### Размещение LRC в сообщении

Когда 8-ми битовое поле LRC (два ASCII символа) передается в сообщении, то старший символ будет передан первым, а за ним - младший. Например, если значение LRC равно 61 hex (0110 0001):

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| ':' | Ад-рес | Функ-ция | Сч-к байт | Байт | Байт | Байт | Байт | LRC Ст. симв. | LRC Мл. симв. | CR | LF |
|  |  |  |  |  |  |  |  | 6 | 1 |  |  |

Пример функции на языке C, реализующей генерацию LRC приведен ниже. Функция принимает два аргумента:

unsigned char \*auchMsg; Указатель на буфер данных

unsigned short usDataLen; Количество байт в буфере

Функция возвращает LRC как тип unsigned char.

ПРИМЕР:

static unsigned char LRC(auchMsg, usDataLen)

unsigned char \*auchMsg;/\* Сообщение над которым \*/

/\* вычисляется LRC \*/

unsigned char usDataLen; /\* Количество байт в сообщении \*/

{

unsigned char uchLRC=0; /\* Инициализация LRC \*/

while(usDataLen)

uchLRC+=\*auchMsg++;

return((unsigned char)(-((char uchLRC)));

}

#### Генерация CRC

CRC это 16-ти разрядная величина, т.е. два байта. CRC вычисляется передающим устройством и добавляется к сообщению. Принимающее устройство также вычисляет CRC в процессе приема и сравнивает вычисленную величину с полем CRC принятого сообщения. Если суммы не совпали - то имеет место ошибка.

16-ти битовый регистр CRC предварительно загружается числом FF hex. Процесс начинается с добавления байтов сообщения к текущему содержимому регистра. Для генерации CRC используются только 8 бит данных. Старт и стоп биты, бит паритета, если он используется, не учитываются в CRC.

В процессе генерации CRC каждый 8-ми битовый символ складывается по ИСКЛЮЧАЮЩЕМУ ИЛИ с содержимым регистра. Результата сдвигается в направлении младшего бита, с заполнением старшего бита нулем. Младший бит извлекается и проверяется. Если младший бит равен 1, то содержимое регистра складывается с определенной ранее, фиксированной величиной, по ИСКЛЮЧАЮЩЕМУ ИЛИ. Если младший бит равен 0, то ИСКЛЮЧАЮЩЕЕ ИЛИ не делается.

Этот процесс повторяется, пока не будет сделано 8 сдвигов. После последнего (восьмого) сдвига, следующий байт складывается с содержимым регистра и процесс повторяется снова. Финальное содержание регистра, после обработки всех байтов сообщения и есть контрольная сумма CRC.

#### Алгоритм генерации CRC:

1. 16-ти битовый регистр загружается числом FF hex (все 1), и используется далее как регистр CRC.
2. Первый байт сообщения складывается по ИСКЛЮЧАЮЩЕМУ ИЛИ с содержимым регистра CRC. Результат помещается в регистр CRC.
3. Регистр CRC сдвигается вправо (в направлении младшего бита) на 1 бит, старший бит заполняется 0.
4. (Если младший бит 0): Повторяется шаг 3 (сдвиг).
5. (Если младший бит 1): Делается операция ИСКЛЮЧАЮЩЕЕ ИЛИ регистра CRC и полиномиального числа 4002 hex.
6. Шаги 3 и 4 повторяются восемь раз.
7. Повторяются шаги со 2 по 5 для следующего сообщения. Это повторяется до тех пор пока все байты сообщения не будут обработаны.
8. Финальное содержание регистра CRC и есть контрольная сумма.

#### Размещение CRC в сообщении

При передаче 16 бит контрольной суммы CRC в сообщении, сначала передается младший байт, затем старший. Например, если CRC равна 1241 hex:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Адрес | Функ-ция | Счетчик байт | Данные | Данные | Данные | Данные | CRC мл. байт | CRC ст. байт |
|  |  |  |  |  |  |  | 41 | 12 |

#### ПРИМЕР

Пример функции на языке C реализующей генерацию CRC приведен ниже. Все возможные величины CRC загружены в два массива. Один массив содержит все 256 возможных комбинаций CRC для старшего байта поля CRC, другой массив содержит данные для младшего байта. Индексация CRC в этом случая обеспечивает быстрое выполнение вычислений новой величины CRC для каждого нового байта из буфера сообщения.

Функция принимает два аргумента:

unsigned char \*puchMsg; /\* Указатель на буфер \*/

unsigned short usDataLen; /\* Количество байтов в буфере \*/

Функция возвращает CRC как тип unsigned short

\_\_flash unsigned char auchCRCHi[256] = {

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,

0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,

0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,

0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81,

0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,

0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,

0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,

0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,

0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01,

0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,

0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,

0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,

0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01,

0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,

0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81,

0x40

};

\_\_flash unsigned char auchCRCLo[256] = {

0x00, 0xc0, 0xc1, 0x01, 0xc3, 0x03, 0x02, 0xc2, 0xc6, 0x06, 0x07, 0xc7, 0x05, 0xc5, 0xc4,

0x04, 0xcc, 0x0c, 0x0d, 0xcd, 0x0f, 0xcf, 0xce, 0x0e, 0x0a, 0xca, 0xcb, 0x0b, 0xc9, 0x09,

0x08, 0xc8, 0xd8, 0x18, 0x19, 0xd9, 0x1b, 0xdb, 0xDA, 0x1A, 0x1E, 0xDE, 0xDF, 0x1F, 0xDD,

0x1D, 0x1C, 0xDC, 0x14, 0xD4, 0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,

0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3, 0xF2, 0x32, 0x36, 0xF6, 0xF7,

0x37, 0xF5, 0x35, 0x34, 0xF4, 0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,

0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29, 0xEB, 0x2B, 0x2A, 0xEA, 0xEE,

0x2E, 0x2F, 0xEF, 0x2D, 0xED, 0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,

0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60, 0x61, 0xA1, 0x63, 0xA3, 0xA2,

0x62, 0x66, 0xA6, 0xA7, 0x67, 0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,

0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68, 0x78, 0xB8, 0xB9, 0x79, 0xBB,

0x7B, 0x7A, 0xBA, 0xBE, 0x7E, 0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,

0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71, 0x70, 0xB0, 0x50, 0x90, 0x91,

0x51, 0x93, 0x53, 0x52, 0x92, 0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,

0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B, 0x99, 0x59, 0x58, 0x98, 0x88,

0x48, 0x49, 0x89, 0x4B, 0x8B, 0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,

0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42, 0x43, 0x83, 0x41, 0x81, 0x80,

0x40

};

unsigned int ModBusCRC(unsigned char \*ByteData, unsigned char NumBytes)

{

unsigned int i, j;

unsigned char CRC16Hi, CRC16Lo;

CRC16Hi = 0xFF;

CRC16Lo = 0xFF;

j =0;

while (NumBytes--)

{

i = CRC16Hi ^ (\*ByteData++);

j++;

CRC16Hi = CRC16Lo ^ auchCRCHi[i];

CRC16Lo = auchCRCLo[i];

}

return ((CRC16Lo << 8) | CRC16Hi);

}

unsigned char CheckModBus(unsigned char \*InBuffAddr, unsigned int LenOutBuff)

{

unsigned int i, j;

if (LenOutBuff <4) return 0;

j = LenOutBuff -2;

i = ModBusCRC(InBuffAddr, j);

InBuffAddr +=j;

j = \*InBuffAddr++;

j += \*InBuffAddr++ <<8;

if (i !=j)

return 0;

return 1;

}

unsigned int FormatModBus(unsigned char \*InBuffAddr, unsigned char \*OutBuffAddr, unsigned int LenInBuff)

{

unsigned int i, j;

unsigned char c;

c = 2;

j = ModBusCRC(InBuffAddr, LenInBuff);

i = LenInBuff;

while (i--)

\*OutBuffAddr++ = \*InBuffAddr++;

\*OutBuffAddr++ = j;

\*OutBuffAddr++ = j >>8;

return ( LenInBuff +c );

}