

Модификации приборов серии 3020. Рекомендации по подключению приборов серии 3020 к компьютеру. Форматы сообщений.

МОДИФИКАЦИИ ПРИБОРОВ СЕРИИ 3020

До 2003 года частотомеры ЕС3020, амперметры ЕА3020 и вольтметры ЕВ3020 выпускались с разъемами типа РП10-11 и версией программного обеспечения «0».

В настоящее время они выпускаются с разъемами типа MSTB 2,5/8-STF и версией программного обеспечения «1».

Отличия в программном обеспечении версии «1» от версии «0» заключаются в следующем (таблица 3):

- изменена функция калибровки, вместо 3 команд - одна;
- в версии «0» только одна скорость обмена 2400 бод, а в версии «1» введена функция установки скорости обмена от 110 до 19200 бод;
- в версии «0» функция «Сброс» сбрасывала прибор в исходное состояние: прибор не калиброван, адрес = 0 и пользовательские данные - пусто, а в версии «1» сбрасывается только регистр состояния;
- изменения в регистре состояния приведены в таблице 4.

В дальнейшем планируется добавить широковежательные команды для одновременного измерения и сохранения в памяти текущего значения (среза), которое может быть прочитано в любое время.

РЕКОМЕНДАЦИИ ПО ПОДКЛЮЧЕНИЮ ПРИБОРОВ СЕРИИ 3020 К КОМПЬЮТЕРУ

1. ПРОТОКОЛ ИНФОРМАЦИОННОГО ОБМЕНА ПРИБОРОВ СЕРИИ 3020

Информационный обмен между сервером (контроллером) и приборами серии 3020 осуществляется посредством 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 приборам кадр имеет размер 8 байт;
- при передаче информации от приборов к серверу – 10 байт.

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

- скорость обмена 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200 бит/с (устанавливается программно);
- число бит данных – 8;
- контроль четности отсутствует;
- число стоп-бит – 1;

Общий формат кадров сервера представлен в таблице 1.

Таблица 1

Номер Байта	Значение	Комментарий
1	10h	Старт- байт
2	Address	Адрес прибора: 0 – используется для калибровки ваттметров и варметров CP3020;
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

Формат кадра ответа приборов серии 3020 на запрос сервера приводится в таблице 2.

Таблица 2

№ Байта	Значение	Комментарий
1	10h	Старт-байт
2	Address	Собственный адрес прибора
3	Function	Повторяет код функции запроса
4	Flags.Low	Младший байт слова состояния прибора

№ Байта	Значение	Комментарий
5	Flags.High	Старший байт слова состояния прибора
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

Приём приборами кадров сервера осуществляется асинхронно и параллельно с внутренними циклами измерения и самодиагностики. Интенсивность запросов сервера не влияет на внутреннюю работу приборов серии 3020.

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

В зависимости от кода функции Function приборы могут игнорировать содержимое полей Mant.Low, Mant.High и EXP в кадре сервера. Значения уставок, коэффициентов трансформации и значения калибровочных отметок, передаются и принимаются в формате с плавающей запятой вида:

$$\text{Число} = \text{Mant} \cdot 2^{\text{EXP}}, \quad (1)$$

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

EXP – экспонента числа – знаковое целое 8-ми разрядное.

Абсолютное значение мантиссы находится в интервале от 16384 до 32768, то есть мантисса выровнена по старшему незнаковому разряду 16-ти разрядного числа. При этом относительная погрешность округления не превышает 0.003%.

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

Коды функций, поддерживаемые приборами, делятся на три категории:

- рабочие функции, используемые непосредственно при эксплуатации приборов в составе телемеханического комплекса;
- функции калибровки;
- функции диагностики.

Назначения кодов функций представлены в таблице 3.

При эксплуатации в составе телемеханического комплекса приборы серии 3020 работают в режиме запрос/ответ. Используются основные рабочие функции «Запрос результата измерения».

При получении кадра с кодом функции «Запрос результата измерения» приборы начинают формировать 10-и байтовый ответ серверу сразу после получения последнего байта в кадре запроса.

Результат измерения, возвращаемый серверу в полях Mant.Low, Mant.High и EXP определяется соотношением (1) и выражен в основных единицах измерения.

Слово состояния, возвращаемое серверу в полях Flags.Low и Flags.High, имеет битовый характер и изменяется во внутреннем цикле самодиагностики приборов. Его формат представлен в таблице 4.

После ответа на запрос сервера приборы вновь настраиваются на приём нового кадра.

Таблица 3

Наименование Функции	ЕВ3020 версия 0	ЕВ3020 версия 1	ЕА3020 версия 0	ЕА3020 версия 1	ЕС3020 версия 0	ЕС3020 версия 1	СР3020 (ваттметр)	СР3020 (варметр)	Примечание
“Запрос результата измерения”	55h	55h	49h	49h	46h	46h			ASCII-код символа “U, I или F”.
	-	-	-	-	-	-	50h, 5Fh	50h, 5Fh	ASCII-код символа “P, _”.
	-	-	-	-	-	-	50h, 61h	50h, 61h	ASCII-код символа “P, a”.
	-	-	-	-	-	-	50h, 62h	50h, 62h	ASCII-код символа “P, b”.
	-	-	-	-	-	-	50h, 63h	50h, 63h	ASCII-код символа “P, c”.
	-	-	-	-	-	-	51h 5Fh	51h 5Fh	ASCII-код символа “Q, _”.
	-	-	-	-	-	-	51h, 61h	51h, 61h	ASCII-код символа “Q, a”.
	-	-	-	-	-	-	51h, 62h	51h, 62h	ASCII-код символа “Q, b”.
	-	-	-	-	-	-	51h, 63h	51h, 63h	ASCII-код символа “Q, c”.
	-	-	-	-	-	-	55h, 61h	55h, 61h	ASCII-код символа “U, a”.
	-	-	-	-	-	-	55h, 62h	55h, 62h	ASCII-код символа “U, b”.
	-	-	-	-	-	-	55h, 63h	55h, 63h	ASCII-код символа “U, c”.
	-	-	-	-	-	-	49h, 61h	49h, 61h	ASCII-код символа “I, a”.

Наименование Функции	ЕВ3020 версия 0	ЕВ3020 версия 1	ЕА3020 версия 0	ЕА3020 версия 1	ЕС3020 версия 0	ЕС3020 версия 1	СР3020 (ваттметр)	СР3020 (варметр)	Примечание
	-	-	-	-	-	-	49h, 62h	49h, 62h	ASCII-код символа "l, b".
	-	-	-	-	-	-	49h, 63h	49h, 63h	ASCII-код символа "l, c".
“Адресация прибора”	80h	80h	80h	80h	80h	80h	80h	80h	
“Передача коэффициента трансформации”	81h	81h	81h	81h	-	-	-	-	
“Передача коэффициента трансформации Кн”	-	-	-	-	-	-	81h	81h	
“Передача коэффициента трансформации Кт”	-	-	-	-	-	-	82h	82h	
“Передача нижней уставки”	82h	82h	82h	82h	82h	82h	-	-	
“Передача верхней уставки”	83h	83h	83h	83h	83h	83h	83h	-	
“Установить скорость передачи”	-	8Dh	-	8Dh	-	8Dh	8Dh	8Dh	Следующий байт [0..8]
“Запись пользовательских данных”	8Eh	8Eh	8Eh	8Eh	8Eh	8Eh	8Eh	8Eh	
“Чтение коэффициента трансформации”	91h	91h	91h	91h	-	-	-	-	

Наименование Функции	ЕВ3020 версия 0	ЕВ3020 версия 1	ЕА3020 версия 0	ЕА3020 версия 1	ЕС3020 версия 0	ЕС3020 версия 1	СР3020 (ваттметр)	СР3020 (варметр)	Примечание
“Чтение коэффициента трансформации Кн”	-	-	-	-	-	-	91h	91h	
“Чтение коэффициента трансформации Кт”	-	-	-	-	-	-	92h	92h	
“Чтение нижней уставки”	92h	92h	92h	92h	92h	92h	-	-	
“Чтение верхней уставки”	93h	93h	93h	93h	93h	93h	93h	93h	
“Чтение пользовательских данных”	9Eh	9Eh	9Eh	9Eh	9Eh	9Eh	9Eh	9Eh	
«Передача нижней калибровочной отметки»	A1h	-	A4h	-	-	-	-	-	
«Передача верхней калибровочной отметки»	A2h	-	A5h	-	-	-	-	-	
«Активизация калибровочной информации»	A3h	-	A6h	-	-	-	-	-	
“Калибровка U, I”	-	A2h	-	A5h	-	-	-	-	
“Калибровка частоты”	-	-	-	-	D1h	D1h	-	-	
Калибровка Ua	-	-	-	-	-	-	A1h	A1h	

Наименование Функции	ЕВ3020 версия 0	ЕВ3020 версия 1	ЕА3020 версия 0	ЕА3020 версия 1	ЕС3020 версия 0	ЕС3020 версия 1	СР3020 (ваттметр)	СР3020 (варметр)	Примечание
Калибровка Ub	-	-	-	-	-	-	B1h	B1h	
Калибровка Uc	-	-	-	-	-	-	C1h	C1h	
Калибровка Ia	-	-	-	-	-	-	A4h	A4h	
Калибровка Ib	-	-	-	-	-	-	B4h	B4h	
Калибровка Ic	-	-	-	-	-	-	C4h	C4h	
“Стохастическое чтение кода выборки АЦП”	E1h	E1h	E1h	E1h	-	-	-	-	
“Стохастическое чтение кода выборки АЦП”	-	-	-	-	-	-	E1h	E1h	Следующий - [1..7] байт соответствует – [Ua,Ub,Uc,Ia,Ib,Ic,Vref]
«Тест EPROM»	FEh	-	FFh	-	FFh	-	FFh	FFh	Сброс флагов ошибок
“Сброс прибора”	FFh	-	FFh	-	FFh	-	-	-	Сброс прибора. Не калиброван, адрес = 0, пользовательские данные – пусто.
“Сброс прибора”	-	FFh	-	FFh	-	FFh	FFh	FFh	Сброс флагов ошибок

Таблица 4

№ бита	EA3020 версия «0»	EA3020 версия «1»	EB3020 версия «0»	EB3020 версия «1»	EC3020 версия «0»	EC3020 версия «1»	CP3020
0	0	Сбой программы	0	Сбой программы	0	Сбой программы	Сбой программы
1	Сбой синхронизации АЦП	Сбой АЦП	Сбой синхронизации АЦП	Сбой АЦП	0	-	Сбой АЦП
2	Сбой опоры АЦП	Сбой опоры АЦП	Сбой опоры АЦП	Сбой опоры АЦП	0	-	Сбой опоры АЦП
3	Переполнение АЦП	Переполнение АЦП	Переполнение АЦП	Переполнение АЦП	0	-	Переполнение АЦП
4	Аппаратный сбой EPROM	Сбой EEPROM	Аппаратный сбой EPROM	Сбой EEPROM	Аппаратный сбой EPROM	Сбой EEPROM	Сбой EEPROM
5	Логический сбой EPROM	-	Логический сбой EPROM	-	Логический сбой EPROM	-	-
6	0	-	0	-	0	-	-
7	0	Сбой генератора	0	Сбой генератора	0	Сбой генератора	Сбой генератора
8	0	-	0	-	0	-	-
9	Калибровка амперметра EA3020 разрешена (была принята информация о калибровочных отметках)	-	Калибровка вольтметра EB3020 разрешена (была принята информация о калибровочных отметках)		0	-	-

№ бита	EA3020 версия «0»	EA3020 версия «1»	EB3020 версия «0»	EB3020 версия «1»	EC3020 версия «0»	EC3020 версия «1»	CP3020
10	Амперметр EA3020 не калиброван (используется калибровочная константа по умолчанию).	-	Вольтметр EB3020 не калиброван (используется калибровочная константа по умолчанию).	-	Частотомер EC3020 не калиброван (используется калибровочная константа по умолчанию).	-	-
11	Амперметр EA3020 не адресован (используется 0-й адрес по умолчанию).	-	Вольтметр EB3020 не адресован (используется 0-й адрес по умолчанию).	-	Частотомер EC3020 не адресован (используется 0-й адрес по умолчанию).	-	-
12	Нижняя уставка	Нижняя уставка	Нижняя уставка	Нижняя уставка	Нижняя уставка	Нижняя уставка	-
13	Верхняя уставка	Верхняя уставка	Верхняя уставка	Верхняя уставка	Верхняя уставка	Верхняя уставка	Верхняя уставка
14	Переполнение (переполнение индикатора или перегрузка АЦП)	-	Переполнение (переполнение индикатора или перегрузка АЦП)	-	Переполнение (превышена максимально допустимая частота)	-	-
15	Результаты измерений не достоверны	Результаты измерений не достоверны	Результаты измерений не достоверны	Результаты измерений не достоверны	Результаты измерений не достоверны	Результаты измерений не достоверны	Результаты измерений не достоверны

Для функции “Адресация прибора” поле Address в кадре сервера содержит текущий адрес прибора. Поле Mant.Low – новый адрес. Содержимое Mant.High и EXP игнорируется. Применение данной функции заставляет реагировать приборы только на новый адрес, сохраняющийся и при отключении питания. Кадр ответа для данной функции не возвращается.

Для функций передачи значения уставок, коэффициентов трансформации значения передается в формате (1). Установленные значения сохраняются при отключении питания приборов. Кадр ответа не возвращается.

Коэффициенты трансформации и уставки устанавливаются независимо. Это означает, что при передаче новых значений коэффициента трансформации текущие уровни уставок не изменяются, при этом изменившиеся показания приборов могут выйти за их границы.

Для функции скорость передачи значение номера скорости (Таблица 5) передается в поле Mant.Low. Установленное значение сохраняется при отключении питания приборов. Кадр ответа не возвращается.

Таблица 5

Значение	0	1	2	3	4	5	6	7	8
бит/с	110	150	300	600	1200	2400	4800	9600	19200

Для функции “Запись пользовательских данных” адрес записываемой ячейки (от 0 до 31) передается в поле Mant.Low кадра сервера, а её содержимое – в поле Mant.High. Поле EXP игнорируется. Пользовательские данные сохраняются при выключении питания. Кадр ответа не требуется. В пользовательских данных потребитель может записать любую информацию длиной до 32 символов, например: место установки прибора, срок очередной поверки и др.

Для функции чтения значения уставок, коэффициентов трансформации его текущее значение возвращается в кадре ответа приборов в формате (1). Приборы формируют ответ так же, как и при запросе результата измерения.

Для функции “Чтение пользовательских данных” адрес читаемой ячейки (от 0 до 31) передается в поле Mant.Low кадра сервера, а её содержимое возвращается в поле Mant.Low кадра ответа приборов. В поле Mant.High возвращается тип прибора (Таблица 6), в поле EXP – версия программы.

Таблица 6

Тип прибора	EA3020	EB3020	EC3020	CP3020 (ваттметр)	CP3020 (варметр)
Возвращаемое значение	49h	55h	46h	50h	51h

Для функций “Калибровки” точные значения калибровочных отметок, выраженные в амперах, вольтах или герцах, приведённые к измерительному входу приборов (без учёта коэффициента трансформации), передаются в кадре сервера в формате (1), после установления показаний приборов на данных отметках. Кадр ответа не предусмотрен. Калибровка возможна только если установлен адрес 0.

Для функции “Стохастическое чтение кодов выборок АЦП” содержимое полей Mant.Low, Mant.High и EXP незначимо и игнорируется. Данная функция используется при настройке прибора и запрашивает кадр ответа прибора, формирующийся так же, как и при запросе результата измерений, но с содержимым поля Mant (два байта) в виде без знакового 12-ти разрядного кода выборки внутреннего АЦП прибора. Поле EXP не используется. Выборки АЦП, полученные с помощью этой функции, за счёт малой скорости интерфейсного обмена и особенностей реализации измерительного алгоритма приборов, носят характер близкий к случайному закону.

Для функции “Сброс прибора” содержимое полей Mant.Low, Mant.High и EXP незначимо и игнорируется. Данная функция сбрасывает только флаги ошибок регистров состояния приборов.

Большинство функций, для которых не требуется ответа, заставляют приборы осуществлять внутренние, относительно медленные операции записи в EEPROM. При посылке кадра сервера с кодом такой функции приборы продолжают осуществлять циклы измерений и самодиагностики, но не реагируют на новые запросы сервера в течение времени примерно 100мсек.

2. РЕКОМЕНДАЦИИ ПО НАПИСАНИЮ ПРОГРАММ

Для подключения приборов серии 3020 к компьютеру могут использоваться адаптеры CX3020 (производства предприятия «ЗИП-Научприбор») или адаптеры сторонних производителей, например: адаптеры серии i7000 с автоматическим переключением на прием-передачу, например:

i-7520R: RS-232 to RS-485 Converter;

i-7561: USB to RS232/422/485 Converter.

Для того чтобы адаптер i-7520R работал с программами, написанными для CX3020, необходимо отключить «контроль соединения», для чего соединить в разьеме интерфейса RS-232 сигналы DTR, DSR и CTS.

В адаптерах CX3020 для переключения на прием-передачу используется интерфейсный сигнал RTS (RS232). Команды управления сигналом RTS выглядят так:

Port[AdrCom+4]:=\$01; - переключить на прием.

Port[AdrCom+4]:=\$03; - переключить на передачу.

Адаптер должен быть всегда включен на прием и переключаться на передачу только на момент передачи сообщения. Конец передачи сообщения необходимо контролировать по освобождению передающего сдвигового регистра UART, следующей командой:

repeat until (Port[AdrCom+5] and \$60)=\$60.

Примечание. UART имеет по два регистра на прием и передачу: буферный и сдвиговый.

Т.к. приборы серии 3020 начинают ответ практически сразу после приема запроса внешнее устройство должно быть готово к приему сразу после выдачи запроса. Поэтому при работе под WINDOWS, из-за переключения между задачами возможны потери отдельных байтов и сообщений в целом. Для нормальной работы под WINDOWS, на момент приема передачи необходимо вводить «монопольный» режим.

3. ПРИМЕРЫ ОТДЕЛЬНЫХ ФУНКЦИЙ ДЛЯ АДАПТЕРА СХ3020

Ниже приведены примеры отдельных функций связи по COM-порту для адаптера СХ3020, написанные на Borland pascal 7.0, для работы под DOS.

function ReadByteCom:byte; - чтение байта из COM-порта, возвращает принятый байт.

procedure WriteByteCom(Data:byte); - передача байта в COM-порт

procedure ReadMsg(Len:byte); - чтение всего сообщения, **ReadBuf[i]** – буфер принятого сообщения.

procedure WriteMsg(Len:byte); - передача всего сообщения, **WriteBuf[i]** – буфер передаваемого сообщения.

function BufToReal:real; - преобразование **Real** CP3020 в **Real** pascal при приеме.

procedure RealToBuf(R:real); - преобразование **Real** pascal в **Real** для передачи.

procedure ReadMeas; - запрос и чтение результатов измерения для ваттметра CP3020.

procedure ReadUserData; - запрос и чтение пользовательских данных, Кн, Кт и уставки.

```
{Используемые глобальные объявления}
```

```
var
```

```
  AdrCom : byte; { физический адрес COM-порта, может принимать значения $3F8 $2F8}
```

```
  AdrInstr : byte; { адрес прибора, может принимать значения 0..255}
```

```
  ErrR_T : byte; { флаги ошибок связи}
```

```
  ReadBuf : array[0..7] of byte;
```

```
  WriteBuf : array[0..9] of byte;
```

```
{-----Процедуры работы с COM портом-----}
```

```
function ReadByteCom:byte;
```

```
{Чтение байта данных из COM-порта/полудуплекс
```

```
  ErrR_T - байт кода завершения процедуры;
```

```
  ErrR_T=0 - все окей
```

```
    7 6 5 4 3 2 1 0
```

```
  | | | | | | | |
```

```
  | | | | | | | - Затирание байта приема
```

```
  | | | | | | - Ошибка четности (шум в линии)
```

```
  | | | | | - Ошибка окружения (передача не синхронизирована)
```

```
  | | | | - Получен перерыв (длинная строка единиц)
```

```
  | | | - Модем не готов передавать байт
```

```
  | | - Модем не включен
```

```
  | - Тайм-Аут}
```

```
var i:word;
```

```
begin
```

```
  i:=0;
```

```
  repeat until (Port[AdrCom+5] and $60)=$60;
```

```
    {Ждём пока не завершится передача}
```

```
  Port[AdrCom+4]:=$01; {Компьютер активен и готов читать}
```

```
  ErrR_T:=(not (Port[AdrCom+6] or $CF)) shl 1; {Готов модем ?}
```

```
  ErrR_T:= ErrR_T or (Port[AdrCom+5] and $1E); {Проверка ошибок}
```

```
  if ErrR_T=0 then begin
```

```
    repeat {цикл ожидания байта, выход если байт не получен}
```

```
      inc(i);
```

```
      if i=0 then break;
```

```

    until (Port[AdrCom+5] and $01)<>0; {Есть байт ?}
    if i=0 then begin {байт не получен}
        ErrR_T:=$80;
        ReadByteCom:=0;
        end
    else ReadByteCom:=Port[AdrCom];
    end;
end;

procedure WriteByteCom(Data:byte);
{Процедура передачи байта Data в COM-порт/полудуплекс
ErrR_T - код завершения процедуры;
ErrR_T=0 - все окей
 7 6 5 4 3 2 1 0
| | | | | | | |
|           |- Ошибка окружения (передача не синхронизирована)
|- Модем не включен}

begin
    ErrR_T:=(not(Port[AdrCom+6] or $0DF)) shl 1; {Готов модем ?}
    ErrR_T:=ErrR_T or (Port[AdrCom+5] and $08);
    if ErrR_T=0 then begin
        repeat until (Port[AdrCom+5] and $20)<>0;
        {Ожидаем освобождения регистра хранения}
        Port[AdrCom]:=Data;
        end;
end;

procedure ReadMsg(Len:byte);
{Процедура приема посылки по COM-порту
ErrR_T - коды завершения процедуры}

var i:byte;

begin
    for i:=0 to Len-1 do begin
        ReadBuf[i]:=ReadByteCom;
        if ErrR_T<>0 then break;
        end;
end;

procedure WriteMsg(Len:byte);
{Процедура посылки по COM-порту
WriteBuf - буфер передачи;
ErrR_T - коды завершения процедуры}

var i:byte;

begin
    Port[AdrCom+4]:=$03; {Компьютер активен и готов к передаче}
    delay(1);
    for i:=0 to Len-1 do begin
        WriteByteCom(WriteBuf[i]);
        if ErrR_T<>0 then break;
        end;
i:=Port[AdrCom]; {Холостое чтение, чтоб приемник отработал}
end;

```

```

function BufToReal:real;
var
  a : integer;
  ab : array[0..1] of byte absolute a;
  R : real;
begin
  ab[0]:=ReadBuf[5];
  ab[1]:=ReadBuf[6];
  { Вариант преобразования через логарифм}
  BufToReal:=a*exp(LN2*ShortInt(ReadBuf[7]));
  { вариант преобразования через цикл
  R:=a;
  while ReadBuf[7]<>0 do
    if ReadBuf[7]>0 then begin
      ReadBuf[7]:=ReadBuf[7]-1;
      R:=R*2;
    end
    else begin          {ReadBuf[7]<0}
      ReadBuf[7]:=ReadBuf[7]+1;
      R:=R/2;
    end;
  BufToReal:=R;}
end;

procedure RealToBuf(R:real);
Var
  Mant:integer;
  Ex:shortInt;
begin
  if R<>0 then begin
    Ex:=15-1-Trunc(ln(abs(R))/LN2);
    Mant:=Round(R*exp(Ex*LN2));
  end
  else begin Mant:=0; Ex:=0; end;
  WriteBuf[3]:=Lo(Mant);
  WriteBuf[4]:=Hi(Mant);
  WriteBuf[5]:=-Ex;
end;

procedure ReadMeas;
var
  i : word;
begin
  if KeyPressed then break;
  WriteBuf[0]:=$10;
  WriteBuf[1]:=AdrInstr;
  WriteBuf[2]:=$50;
  WriteBuf[3]:=$5F;
  WriteBuf[6]:=0;
  for i:=1 to 5 do WriteBuf[6]:=WriteBuf[6]+WriteBuf[i];
  WriteBuf[7]:=$16;
  WriteMsg(8);
  ReadMsg(10);
  if (ErrR_T=0) and TestBufRXD then BufMeas[i]:=BufToReal
  OutMeas; {Вывод значения на экран}
end;

```



```

procedure ReadUserData;
var
  Str:string[32];
  i,j:byte;
begin
  Str:='';
  for j:=1 to 32 do begin
    if KeyPressed then break;
    WriteBuf[0]:=$10;
    WriteBuf[1]:=AdrInstr;
    WriteBuf[2]:=$9E;
    WriteBuf[3]:=j-1;
    WriteBuf[6]:=0;
    for i:=1 to 5 do WriteBuf[6]:=WriteBuf[6]+WriteBuf[i];
    WriteBuf[7]:=$16;
    WriteMsg(8);
    ReadMsg(10);
    if (ErrR_T=0) and TestBufRXD then Str:=Str+char(ReadBuf[5])
    else Str:=Str+' ';
    end;
  win4;
  gotoXY(1,1); write(Str); {Вывод на экран пользовательских
данных}
  {КН}
  WriteBuf[0]:=$10;
  WriteBuf[1]:=AdrInstr;
  WriteBuf[2]:=$91;
  WriteBuf[6]:=0;
  for i:=1 to 5 do WriteBuf[6]:=WriteBuf[6]+WriteBuf[i];
  WriteBuf[7]:=$16;
  WriteMsg(8);
  ReadMsg(10);
  if (ErrR_T=0) and TestBufRXD then begin
    gotoXY(35,1);
    write('КН = ',BufToReal:6:1);
    end;
  {КТ}
  WriteBuf[0]:=$10;
  WriteBuf[1]:=AdrInstr;
  WriteBuf[2]:=$92;
  WriteBuf[6]:=0;
  for i:=1 to 5 do WriteBuf[6]:=WriteBuf[6]+WriteBuf[i];
  WriteBuf[7]:=$16;
  WriteMsg(8);
  ReadMsg(10);
  if (ErrR_T=0) and TestBufRXD then begin
    gotoXY(48,1);
    write('КТ = ',BufToReal:6:1);
    end;
  {уставка}
  WriteBuf[0]:=$10;
  WriteBuf[1]:=AdrInstr;
  WriteBuf[2]:=$93;
  WriteBuf[6]:=0;
  for i:=1 to 5 do WriteBuf[6]:=WriteBuf[6]+WriteBuf[i];
  WriteBuf[7]:=$16;

```

```
WriteMsg(8);
ReadMsg(10);
if (ErrR_T=0) and TestBufRXD then begin
  gotoXY(60,1);
  write('Уст. = ',BufToReal:6:1);
end;
end;
```