

Ethernet модуль Jerome

ТСР/IP команды управления



Версия 1.05

28 Марта 2014

История документа:

Версия	Дата	Описание
1.5	28 Марта 2014	Отображено появление новой версии “прошивки” модуля Jm07, в которой поддерживается ряд новых команд управления: - \$KE,UDT - \$KE,CAT - \$KE,CAC
1.04	17 Октября 2012	Отображено появление новой версии “прошивки” модуля Jm05, в которой поддерживается ряд новых команд управления: - \$KE,SPB - \$KE,MSK - \$KE,SAV - \$KE,GTW - \$KE,SEC - поддержка ряда команд управления последовательным портом
1.03	23 Декабря 2011	Добавлена секция команд управления для последовательного порта модуля (USART)
1.02	01 Июня 2011	Добавлены следующие новые команды: <ul style="list-style-type: none"> • \$KE,PSW • \$KE,MAC
1.01	11 Апреля 2011	Исходная версия документа

Содержание

Версия модуля.....	4
Введение	5
\$KE	6
\$KE,WR.....	6
\$KE,WRA.....	7
\$KE,RD.....	9
\$KE,RID	10
\$KE,ADC.....	11
\$KE,IMPL.....	12
\$KE,PWM,SET.....	13
\$KE,PWM,GET.....	14
\$KE,PFR,SET.....	14
\$KE,PFR,GET	15
\$KE,SPB,SET	16
\$KE,SPB,GET	17
\$KE,IO,SET	17
\$KE,IO,GET	18
\$KE,DAT	19
\$KE,EVT	20
\$KE,PSW,SET	21
\$KE,PSW,NEW	21
\$KE,SEC,SET.....	22
\$KE,SEC,GET	23
\$KE,SAV,SET	23
\$KE,SAV,FLS	24
\$KE,SAV,GET	24
\$KE,IP,SET.....	25
\$KE,IP,GET.....	25
\$KE,MAC,SET.....	26
\$KE,MAC,GET	26
\$KE,MSK,SET	27
\$KE,MSK,GET.....	28
\$KE,GTW,SET.....	28
\$KE,GTW,GET	29
\$KE,UDT,SET.....	29
\$KE,UDT,GET.....	29
\$KE,CAT,SET	30
\$KE,CAT,ON/OFF.....	33
\$KE,CAC.....	34
\$KE,INF.....	34
\$KE,RST.....	34
\$KE,DEFAULT	35

Версия модуля



Данная редакция документа соответствует модулю Jerome со следующими характеристиками:

Версия программного обеспечения ("прошивка")	Jm07
Версия Web-интерфейса	JW7.0
Версия платы:	Rev.D

Введение

Для управления модулем Jerome предназначен набор команд в текстовом формате, называемых KE командами. Для управления модулем с помощью KE-команд необходимо установить TCP/IP сетевое соединение с адресом 192.168.0.101 (по умолчанию) по порту 2424. После успешного установления соединения можно отправлять управляющие команды и получать ответы модуля.

В качестве программного обеспечения можно использовать любую терминальную программу позволяющую устанавливать сетевое соединение по протоколу TCP/IP, например программу *HyperTerminal*, по умолчанию входящую в состав ОС Windows XP.

Для защиты модуля от несанкционированного доступа в нем реализована система контроля доступа с помощью пароля. Модуль не выполняет команды управления до тех пор, пока не будет введен корректный пароль.

Любая KE команда, отсылаемая модулю, должна начинаться с символов '\$KE'. Также все команды должны заканчиваться символом возврата каретки <CR> и символом перехода на новую строку <LF> (в шестнадцатеричном формате эти символы имеют коды 0x0D и 0x0A соответственно).

\$KE,Команда<CR><LF>

Ответы модуля на команды, а также отдельные информационные блоки выдаваемые модулем всегда начинаются с символа '#' (шестнадцатеричный код 0x23) и заканчиваются символами возврата каретки <CR> и перехода на новую строку <LF>.

#Ответ модуля<CR><LF>

Далее по тексту документа символы <CR><LF>, которыми должна заканчиваться любая команда модулю и любой ответ выдаваемый модулем, опускаются.

В том случае, если, синтаксис команды, отправленной модулю, не является верным, модуль выдает сообщение об ошибке:

#ERR

В составе модуля Jerome встроен последовательный порт USART. Помимо передачи данных по сетевому интерфейсу (TCP-COM интерфейс), модуль также может принимать и обрабатывать ряд текстовых команд управления через последовательный порт. Общие правила построения команд управления для последовательного порта ни чем не отличаются от правил и принципов, описанных выше. Скорость порта по умолчанию равна 9600 бит/с, но может быть изменена. Те команды, которые поддерживаются не только на уровне командного TCP порта, но и последовательным портом USART помечены специальным символом:



- команда поддерживается через порт USART

\$KE

Команда проверки работоспособности модуля. Это простая тестовая команда, на которую модуль должен ответить '#OK'.

Синтаксис: \$KE

Ответ на запрос:

#OK

Пример:



Тестовая проверка модуля:

запрос: \$KE
ответ: #OK



команда поддерживается через порт USART

\$KE,WR

Синтаксис (Вариант 1): \$KE,WR,<LineNumber>,<Value>

С помощью данной команды можно установить высокий ($Value = 1$) или низкий уровень напряжения ($Value = 0$) на линии ввода / вывода модуля под номером *LineNumber*. Линия *LineNumber* должна быть настроена на выход (по умолчанию линии ввода / вывода настроены на выход. Изменение направления линии производится с помощью команды \$KE,IO,SET).

Параметры:

LineNumber – номер линии ввода/вывода. Может быть в пределах от 1 до 22 включительно.

Value – значение для записи на линию. 1 – высокий уровень напряжения (+3.3 В), 0 – низкий уровень напряжения (0 В).

Ответ на запрос:

#WR,OK – значение успешно установлено.

#WR,WRONGLINE – команда сформирована верно, но линия *LineNumber* сконфигурирована на вход, и значение не может быть установлено.

Пример:



Установим высокий уровень напряжения на дискретной линии ввода/вывода под номером 16:

запрос: \$KE,WR,16,1
ответ: #WR,OK

Синтаксис (Вариант 2): \$KE,WR,ALL,<State>

С помощью данной команды можно установить высокий или низкий уровень напряжения на всех дискретных линиях (настроенных на выход) одновременно.

Параметры:

State – если равен *ON* – на всех линиях будет выставлен высокий уровень, *OFF* – соответственно низкий.

Ответ на запрос:

#WR,OK – значение успешно установлено.

Пример:



Установим высокий уровень напряжения на всех дискретных линиях модуля, настроенных на выход:

запрос: \$KE,WR,ALL,ON
ответ: #WR,OK

Примечание:



Установленное значение может быть сохранено в энергонезависимой памяти и автоматически применено после сброса питания, если активирована команда \$KE,SAV.

\$KE,WRA

Команда \$KE,WRA позволяет за одно обращение установить произвольную комбинацию высоких или низких уровней напряжения на всех дискретных линиях модуля, настроенных на выход.

Синтаксис: \$KE,WRA,<ArrayOfValues>

Параметры:

ArrayOfValues – строка длиной от 1 до 22 символов. Может содержать символы '0' (низкий уровень), '1' (высокий уровень) или 'x' (пропустить линию). Нумерация символов в строке производится слева на право. Значение первого символа строки будет установлено на линии вывода номер 1, значение второго символа - на линии номер 2 и т.д. Если соответствующая линия настроена на вход, то операция записи производится не будет. Строка может содержать

меньшее число символов, чем суммарное число выходных линий, например, строка из 4-х символов позволит установить значение на первых четырех линиях.

Ответ на запрос:

#WRA,OK,<UpdCount> – где *UpdCount* содержит количество успешно записанных значений.

Пример 1:



Считаем, что все линии модуля настроены на выход. Установим на всех линиях логическую единицу, кроме линии под номером 22 для которой установим логический ноль:

запрос: \$KE,WRA,11111111111111111110
ответ: #WRA,OK,22

Пример 2:



Установим на линиях 3 и 12 логическую единицу, остальные линии оставим без изменения:

запрос: \$KE,WRA,xx1xxxxxxxxx1xxxxxxxxxxx
ответ: #WRA,OK,2

Пример 3:



Считаем, что все линии модуля настроены на выход кроме линий номер 2, 3 и 10 которые соответственно настроены на вход. Установим на всех линиях логическую единицу:

запрос: \$KE,WRA,11111111111111111111
ответ: #WRA,OK,19

Входные линии действием этой команды затронуты не будут. При этом значения на всех выходных линиях будут обновлены. Как видно, ответ содержит информацию о том, что были успешно обновлены только 19 линий.

Пример 4:



Установим на первых 8-ми дискретных линиях модуля логический ноль:

запрос: \$KE,WRA,00000000
ответ: #WRA,OK,8

Примечание:



Установленное значение может быть сохранено в энергонезависимой памяти и автоматически применено после сброса питания, если активирована команда \$KE,SAV.

\$KE,RD**Синтаксис (Вариант 1): \$KE,RD,<LineNumber>**

С помощью данной команды можно считать информацию с линии ввода/вывода *LineNumber*. Линия должна быть настроена на вход.

Параметры:

LineNumber – номер линии ввода/вывода. Может быть в пределах от 1 до 22 включительно.

Ответ на запрос:

#RD,<LineNumber>,<Value> – чтение линии *LineNumber* произведено успешно, результат *Value*. *Value = 0* – на вход линии установлен низкий уровень напряжения, *Value = 1* – соответственно, высокий уровень напряжения.

#RD,WRONGLINE – команда сформирована верно, но линия *LineNumber* сконфигурирована на выход, и значение не может быть считано.

Пример:

Считаем текущее состояние линии 2 (настроенной на вход):

запрос: \$KE,RD,2
ответ: #RD,02,1

Синтаксис (Вариант 2): \$KE,RD,ALL

По данной команде модуль произведет последовательный перебор всех линий ввода/вывода. Если линия настроена на вход будет произведено считывание информации с этой линии. Если линия настроена на выход, считывание производиться не будет. Результат выводится в виде сводной строки данных, состоящей из 22 символов (по суммарному числу линий ввода/вывода). Нумерация позиции символа в строке осуществляется слева на право и соответствует номеру линии. Если линия под номером *N* настроена на вход, то символ в сводной строке под номером *N* будет содержать результат чтения. Если линия настроена на выход – в соответствующей позиции строки будет выставлен символ 'x'.

Ответ на запрос:

#RD,<Line1 Value>< Line2 Value>< Line3 Value>.... <Line22 Value>

Пример:

Считать информацию со всех входных дискретных линий модуля:

запрос: \$KE,RD,ALL
ответ: #RD,xxx10xxx0xxx1xxxxxxxxxx

Данный пример показывает, что линии под номером 4, 5, 9 и 13 настроены на вход и на них установлены логические уровни 1, 0, 0, 1 соответственно. Остальные линии настроены на выход.

\$KE,RID

Синтаксис (Вариант 1): `$KE,RID,<LineNumber>`

С помощью данной команды можно считать информацию с линии ввода/вывода *LineNumber*. Линия может быть настроена как на вход, так и на выход. Если линия настроена на вход, то команда возвращает значение логического уровня (0 или 1) установленного на линии “снаружи” модуля, т.е. действие команды полностью аналогично команде \$KE,RD. Если линия настроена на выход, то команда вернет последнее значение, которое было установлено для этой линии с помощью команд \$KE,WR или \$KE,WRA.

Параметры:

LineNumber – номер линии ввода/вывода. Может быть в пределах от 1 до 22 включительно.

Ответ на запрос:

`#RID,<LineNumber>,<Value>` – чтение линии *LineNumber* произведено успешно, результат *Value*. *Value = 0* – на линии установлен низкий логический уровень, *Value = 1* – соответственно, высокий логический уровень.

Пример:



Считаем значение с выходной линии модуля, предварительно записав на нее различные значения:

```
запрос:  $KE,WR,5,1
ответ:   #WR,OK
запрос:  $KE,RID,5
ответ:   #RID,05,1
```

```
запрос:  $KE,WR,5,0
ответ:   #WR,OK
запрос:  $KE,RID,5
ответ:   #RID,05,0
```

Пример:



Считать информацию с входной линии номер 2:

```
запрос:  $KE,RID,2
ответ:   #RID,02,1
```

Синтаксис (Вариант 2): \$KE,RID,<Type>

С помощью данной команды можно считать информацию со всех линий (как ввода, так и вывода), только с входных линий или только с выходных. Выбор линий осуществляется параметром *Type*.

Параметры:

Type – выбор линий для чтения. *ALL* – чтение всех линий, *IN* – чтение только входных линий, *OUT* – только выходных.

Ответ на запрос:

#RID,<Type>,<Line1 Value>< Line2 Value>< Line3 Value>.... <Line22 Value>

Ответ за запрос содержит информацию по всем 22 линиям в виде сводной строки данных. Нумерация в строке производится слева на право. Первому символу в строке соответствует линия номер 1, второму символу линия номер 2 и т.д. *Line Value = 0* – на линии установлен низкий логический уровень, *Line Value = 1* – соответственно, высокий логический уровень. Если направление линии не соответствует запрошенному, соответствующий символ в ответной сводной строке будет установлен в значение 'x'.

Пример:

Считать информацию со всех линий модуля:

```
запрос: $KE,RID,ALL
ответ:  #RID,ALL,0001011100111110011111

запрос: $KE,RID,IN
ответ:  #RID,IN,xxx10xxx0xxx1xxxxx1111

запрос: $KE,RID,OUT
ответ:  #RID,OUT,000xx111x011x11001xxxx
```

Данный пример показывает, что линии под номером 4, 5, 9, 13, 19-22 настроены на вход и на них установлены логические уровни 1, 0, 0, 1, 1, 1, 1, 1 соответственно. Остальные линии настроены на выход.

\$KE,ADC**Синтаксис (Вариант 1): \$KE,ADC,<ChannelNumber>**

Считывание результата АЦП с канала модуля под номером *ChannelNumber*.

Параметры:

ChannelNumber – номер канала АЦП. Может быть в пределах от 1 до 4 включительно.

Ответ на запрос:

#ADC,<ChannelNumber>,<Value> – аналоговому напряжению на входе *ChannelNumber* канала АЦП модуля соответствует число *Value*. Для получения величины напряжения входного сигнала в вольтах необходимо воспользоваться следующей формулой:

$$V_{in} = \frac{Value}{1023} \cdot 3.3 [B]$$

Пример:

Получить результат преобразования АЦП с 3-го канала:

запрос: \$KE,ADC,3
ответ: #ADC,3,0645

В данном примере на входе АЦП в момент подачи запроса было напряжение:

$$V_{in} = \frac{645}{1023} \cdot 3.3 = 2.08 [B]$$

Синтаксис (Вариант 2): \$KE,ADC,ALL

Считывание результатов со всех АЦП каналов модуля за один запрос.

Ответ на запрос:

#ADC,ALL,<Channel1_Value>,<Channel2_Value>,<Channel3_Value>,<Channel4_Value>

\$KE,IMPL**Синтаксис (Вариант 1): \$KE,IMPL,<ImplChannel>**

Считывание значения счетчика импульсов под номером *ImplChannel*.

Ответ на запрос:

#IMPL,<ImplChannel>,T,<SystemTime>,I,<Cycle>,<Value>

Параметры:

- SystemTime* – текущее системное время модуля в секундах
- ImplChannel* – номер счетчика импульсов. Может принимать значения от 1 до 4 включительно.
- Cycle* – число циклов. Один цикл равен 32766 импульсов

Value – значение счетчика импульсов, целое число в диапазоне 0 – 32766.

Пример:



Запрос значения счетчика импульсов под номером 3:

запрос: \$KE,IMPL,3
ответ: #IMPL,3,T,1208,2,3612

Данный пример показывает что в момент времени 1208 счетчик импульсов под номером 3 сработал 2 раза по 32766 (2 цикла) и еще 3612 раз. Итого счетчик суммарно сработал: $32766 \times 2 + 3612 = 69144$ раз.

Синтаксис (Вариант 2): \$KE,IMPL,ALL

Считывание значений со всех счетчиков импульсов модуля за один запрос. По этой команде модуль выдаст информацию по каждому счетчику отдельным ответом.

Синтаксис (Вариант 3): \$KE,IMPL,RST

Обнуление значения всех счетчиков импульсов.

Пример:



Произведем обнуление значений счетчиков импульсов:

запрос: \$KE,IMPL,RST
ответ: #IMPL,RST,OK



Значения счетчика импульсов могут быть сохранены в энергонезависимой памяти и автоматически восстановлены после сброса питания, если активирована команда \$KE,SAV.

\$KE,PWM,SET

Управление ШИМ выходом модуля. Команда задает выходную мощность ШИМ сигнала.

Синтаксис: \$KE,PWM,SET,<PowerValue>

Параметры:

PowerValue – параметр, задающий выходную мощность сигнала на ШИМ выходе. Может принимать значения от 0 до 100. При значении равном 100 – ШИМ сигнал имеет 100% теоретическую мощность и 0% при значении равном 0.

Ответ на запрос:

#PWM,SET,OK

Пример:



Установить 60% уровень мощности ШИМ сигнала:

запрос: \$KE,PWM,SET,60

ответ: #PWM,SET,OK



Значение ШИМ может быть сохранено в энергонезависимой памяти и автоматически восстановлено после сброса питания, если активирована команда \$KE,SAV.

\$KE,PWM,GET

Возвращает текущее значение мощности ШИМ сигнала.

Синтаксис: \$KE,PWM,GET

Ответ на запрос:

#PWM,<PowerValue>

Параметры:

PowerValue – выходная мощность сигнала на ШИМ выходе. Может принимать значения от 0% до 100% включительно.

Пример:



Получить значение мощности ШИМ сигнала на текущий момент времени:

запрос: \$KE,PWM,GET

ответ: #PWM,60

\$KE,PFR,SET

Команда позволяет изменять частоту ШИМ сигнала. Установленное значение сохраняется в энергонезависимой памяти.

Синтаксис: \$KE,PFR,SET,<Value>

Параметры:

Value – безразмерная величина, задающая частоту ШИМ сигнала. Может принимать значения от 2 до 255 включительно. Связь параметра *Value* и частоты ШИМ сигнала описывается приближенной формулой ниже:

$$f_{pwm} = \frac{651.042}{Value + 1} \text{ [кГц]}$$

Оценки величины частоты ШИМ сигнала для ряда конкретных значений параметра *Value* представлены в таблице ниже:

Значение Value	Частота ШИМ, кГц
2	217.014
5	108.507
50	12.765
100	6.446
200	3.239
255	2.543

Ответ на запрос:

#PFR,SET,OK

Пример:

Установить максимально возможную частоту ШИМ сигнала:

запрос: \$KE,PFR,SET,2
ответ: #PFR,SET,OK

\$KE,PFR,GET

Запрос текущего значения частоты ШИМ сигнала.

Синтаксис: \$KE,PFR,GET

Ответ на запрос:

#PFR,<Value>

Параметры:

Value – безразмерная величина, задающая частоту ШИМ сигнала. См. подробности в описании команды \$KE,PFR,SET.

Пример:

Запросим текущее значение частоты ШИМ сигнала:

запрос: \$KE,PFR,GET
ответ: #PFR,156

Используя формулу выше, можно убедиться, что частота ШИМ сигнала на данный момент приближенно равна 4.147 кГц.

\$KE,SPB,SET

Команда позволяет изменять скорость последовательно порта модуля. Настройка сохраняется в энергонезависимой памяти модуля.

Синтаксис: \$KE,SPB,SET,<Value>

Параметры:

Value – безразмерная величина, задающая скорость последовательного порта. Может принимать значения от 1 до 7 включительно. Связь параметра *Value* и скорости последовательного порта представлена в таблице ниже:

Значение Value	Скорость порта, бит/с
1	2400
2	4800
3	9600
4	19200
5	38400
6	57600
7	115200

Ответ на запрос:

#SPB,SET,OK

Пример:

Установим скорость последовательно порта 19200 бит/с:

запрос: \$KE,SPB,SET,4
ответ: #SPB,SET,OK

\$KE,SPB,GET

Запрос текущего значения скорости последовательного порта модуля.

Синтаксис: \$KE,SPB,GET

Ответ на запрос:

#SPB,<Value>

Параметры:

Value – безразмерная величина, задающая скорость последовательного порта. См. подробности в описании команды \$KE,SPB,GET.

Пример:



Запросим текущее значение скорости последовательного порта модуля:

запрос: \$KE,SPB,GET
ответ: #SPB,7

Пример показывает, что текущая скорость порта составляет 115200 бит/с.

\$KE,IO,SET

Команда позволяет настроить линию ввода/вывода *LineNumber* в качестве выходной (*IoDirection* = 0) или входной (*IoDirection* = 1). Настройка сохраняется в энергонезависимой памяти модуля. При последующих включениях модуля линия *LineNumber* будет автоматически сконфигурирована в ранее установленное направление.

Синтаксис (Вариант 1): \$KE,IO,SET,<LineNumber>,<IoDirection>

Параметры:

LineNumber – номер линии ввода/вывода. Может быть в пределах от 1 до 22 включительно.

IoDirection – направление ввода/вывода. 1 – (input) на вход, 0 – (output) на выход.

Ответ на запрос:

#IO,SET,OK

Пример:



Установить дискретную линию ввода/вывода под номером 5 в качестве выходной:

запрос: \$KE,IO,SET,5,0
ответ: #IO,SET,OK

Пример:



Установить дискретную линию ввода/вывода под номером 3 в качестве входной:

запрос: \$KE,IO,SET,3,1
ответ: #IO,SET,OK

Синтаксис (Вариант 2): \$KE,IO,SET,ALL,OUT

Позволяет настроить все дискретные линии на выход.

Ответ на запрос:

#IO,SET,OK

Синтаксис (Вариант 3): \$KE,IO,SET,ALL,IN

Позволяет настроить все дискретные линии на вход.

Ответ на запрос:

#IO,SET,OK

\$KE,IO,GET

Синтаксис (Вариант 1): \$KE,IO,GET,ALL

Вывод значений направления ввода/вывода для всех 22 линий модуля в виде сводной строки за один запрос.

Ответ на запрос:

#IO,ALL,<Line1 Direction><Line2 Direction>...<Line22 Direction>

Line Direction = 1 – линия настроена на вход

Line Direction = 0 – линия настроена на выход

Пример:



Получить значения направлений ввода/вывода линий модуля:

запрос: \$KE,IO,GET,ALL
 ответ: #IO,ALL,0001000011000000000000

В данный момент времени линии с номерами 4, 9 и 10 настроены на вход, остальные на выход.

Синтаксис (Вариант 2): \$KE,IO,GET,<LineNumber>

Вывод значения направления ввода/вывода для линии *LineNumber*.

Параметры:

LineNumber – номер линии ввода/вывода. Может быть в пределах от 1 до 22 включительно.

Ответ на запрос:

#IO,<LineNumber>,<Value>
Value – направление ввода/вывода, 0 – выход, 1 – вход.

Пример:



Получить значение направления ввода/вывода линий модуля 13:

запрос: \$KE,IO,GET,13
 ответ: #IO,13,1

Пример показывает, что линия настроена на вход.

\$KE,DAT

Команда включает/выключает выдачу сводной информации по аппаратным ресурсам модуля с частотой 1 Гц. Выводится следующая информация: текущее системное время, значения всех входных дискретных линий, всех выходных линий, значения со всех каналов АЦП и значения счетчиков импульсов.

Синтаксис: \$KE,DAT,<Sate>

Параметры:

Sate – если равен *ON* – производится включение выдачи сводной информации, *OFF* – выдача информации соответственно выключается.

Ответ на запрос:

#DAT,OK

Пример:



Включить периодическую выдачу сводной информации по аппаратным ресурсам:

```
запрос: $KE,DAT,ON
ответ:  #DAT,OK
        #TIME,614
        #RID,IN,0xxxx0xxxxxx0xxxx0xxxx
        #RID,OUT,x0000x000000x0000x0000
        #ADC,ALL,610,529,514,606
        #INT,ALL,614,29,0,0,0
        #IMPL,1,T,614,2,3612
        #IMPL,2,T,614,0,0
        #IMPL,3,T,614,0,0
        #IMPL,4,T,614,0,27519
        #TIME,615
        .....
```

Информация выводится с частотой в 1 Гц.

\$KE,EVT

Команда включает/выключает режим автоматического отслеживания изменения состояний входных дискретных линий (система “Сторож”). Если такой режим включен и на любой из входных линий происходит изменение состояния, в автоматическом режиме производится выдача информационного сообщения об обнаруженном событии. Настройка сохраняется в энергонезависимой памяти.

Синтаксис: **\$KE,EVT,<Sate>**

Параметры:

Sate – если равен *ON* – режим включен, *OFF* – режим выключается.

Ответ на запрос:

```
#EVT,OK
```

Пример:



Включить режим отслеживания изменений на входных линиях:

```
запрос: $KE,EVT,ON
ответ:  #EVT,OK
```

Например, в некий момент времени произошло изменение состояния входной линии под номером 4. Новое состояние – логическая единица. При этом в порт будет выдано сообщение в следующем формате:

```
#EVT,IN,<SystemTime>,<LineNumber>,<CurrentValue>
```

где

SystemTime – текущее системное время модуля в секундах.

LineNumber – номер входной дискретной линии, на которой было обнаружено событие

CurrentValue – текущее значение на входной линии

В описываемом примере ответ может быть таким:

```
#EVT,IN,567,4,1
```

\$KE,PSW,SET

С помощью команды можно ввести пароль доступа к командному интересу модуля (TCP порт 2424). Также эта команда деблокирует передачу данных по TCP порту 2525 (интерфейс TCP – COM).

Синтаксис: `$KE,PSW,SET,<Password>`

Параметры:

Password – Пароль для доступа к модулю

Ответ на запрос:

#PSW,SET,OK – команда сформирована верно, пароль верный, доступ к командному интерфейсу разблокирован
 \$PSW,SET,BAD – неверный пароль. Доступ по-прежнему заблокирован

Пример:



Введем пароль доступа к модулю (по умолчанию - Jerome):

запрос: `$KE,PSW,SET,Jerome`

ответ: `#PSW,SET,OK`



- команда поддерживается последовательным портом

\$KE,PSW,NEW

С помощью этой команды можно установить новый пароль, который будет использоваться для разблокировки доступа к командному интерфейсу (TCP порт 2424) и в качестве пароля доступа к Web-интерфейсу. Новый пароль сохраняется в энергонезависимой памяти.

Синтаксис: `$KE,PSW,NEW,<CurrPassword>,<NewPassword>`

Параметры:

CurrPassword – Текущий пароль доступа
NewPassword – Новый пароль, длиной не более 9 символов

Ответ на запрос:

#PSW,NEW,OK – новый пароль успешно установлен
 \$PSW,NEW,BAD – текущий пароль указан неверно

Пример:

Установить новый пароль “SimSim” (при условии, что текущий пароль соответствует паролю по умолчанию – “Jerome”):

запрос: \$KE,PSW,NEW,Jerome,SimSim
 ответ: #PSW,NEW,OK



В том случае, если вы забыли новый пароль или произошел сбой во время его записи в энергонезависимую память (отключение питания) – единственный выход из сложившейся ситуации является аппаратный сброс настроек. Для сброса всех настроек в энергонезависимой памяти модуля в исходное значение по умолчанию необходимо подать команду \$KE,DEFAULT через последовательный порт модуля или воспользоваться линией 36 (RST) для аппаратного сброса настроек.

\$KE,SEC,SET

Команда задает общую политику безопасности модуля. Она позволяет отключить любые запросы паролей для доступа к модулю (полезно в случае “безопасной” локальной сети, например, при прямом соединении модуля и компьютера). Настройка сохраняется в энергонезависимой памяти модуля.

Синтаксис: \$KE,SEC,SET,<State>

Параметры:

Sate – Если он равен *ON* (значение по умолчанию), то доступ к командному порту TCP 2424, TCP-USART порту 2525 и Web-интерфейсу защищается паролем (пользователь должен указать пароль для входа в интерфейс). Заблокирована возможность управления модулем прямыми HTTP запросами. Если параметр равен *OFF* – то пароли доступа не запрашиваются.

Ответ на запрос:

#SEC,OK

Пример:

Отключим запрос всех паролей для доступа к модулю:

запрос: \$KE,SEC,SET,OFF
ответ: #SEC,OK

\$KE,SEC,GET

Запрос состояния политики безопасности модуля.

Синтаксис: \$KE,SEC,GET

Ответ на запрос:

#SEC,<State>

Параметры:

State – если равен *ON* – доступ к модулю защищен паролем, *OFF* – доступ к модулю полностью разблокирован.



команда поддерживается через порт RS-232

\$KE,SAV,SET

Команда блокирует / деблокирует возможность сохранения состояний аппаратных ресурсов в энергонезависимой памяти и их последующего восстановления и применения после сброса питания. Состояния (значения) следующих аппаратных ресурсов могут быть сохранены и восстановлены после сброса питания:

- выходные дискретные линии
- счетчики импульсов
- ШИМ

Для экономии ресурсов памяти модуля, сохранение состояний производится не мгновенно по факту изменения аппаратного ресурса, а на периодическом базисе каждые 30 секунд. В том случае если сохранить значения необходимо немедленно (например, перед отключением питания), следует использовать команду \$KE,SAV,FLS

Синтаксис: \$KE,SAV,SET,<State>

Параметры:

State – если равен *ON* – режим сохранения включен, *OFF* – выключен.

Ответ на запрос:

#SAV,OK

Пример:



Рассмотрим практический пример:

Включаем режим сохранения состояний:

`$KE,SAV,SET,ON`

Ответ модуля:

`#SAV,OK`

Установим высокий уровень напряжения на дискретной линии под номером 1 (считаем, что она настроена на выход):

`$KE,WR,1,1`

Ожидаем не менее 30 сек и отключаем питание...

Включаем питание. Запрашиваем состояние линии:

`$KE,RID,1`

На линии будет установлен высокий логический уровень:

`#RID,1,1`

\$KE,SAV,FLS

В принудительном порядке сохраняет значения аппаратных ресурсов в энергонезависимую память модуля.

Синтаксис: `$KE,SAV,FLS`

Ответ на запрос:

`#SAV,FLS,OK`

\$KE,SAV,GET

Возвращает текущее состояние режима сохранения значений аппаратных ресурсов в энергонезависимой памяти.

Синтаксис: `$KE,SAV,GET`

Ответ на запрос:

`#SAV,<State>`

Параметры:

State – если равен *ON* – режим включен, *OFF* – выключен.

\$KE,IP,SET

Команда позволяет установить IP адрес модуля. По умолчанию, IP адрес модуля равен 192.168.0.101. Параметр сохраняется в энергонезависимой памяти. Изменения вступают в силу после перезагрузки модуля (команда \$KE,RST или сброс питания).

Синтаксис: `$KE,IP,SET,<IpAddress>`

Параметры:

IpAddress – IP адрес в формате X.X.X.X (в качестве X могут быть использованы числа от 0 до 255). Адреса 0.0.0.0 и 255.255.255.255 запрещены к использованию.

Ответ на запрос:

#IP,SET,OK

Пример:



Установить IP адрес модуля равным 192.168.0.115:

запрос: `$KE,IP,SET,192.168.0.115`

ответ: `#IP,SET,OK`



Будьте внимательны при изменении сетевых настроек модуля. Если адрес будет указан некорректно, вы не сможете подключиться к модулю через сетевое соединение. Для сброса/изменения настроек в этом случае следует использовать линию модуля 36 (RST) или последовательный порт.



- команда поддерживается последовательным портом

\$KE,IP,GET

Возвращает текущий IP адрес модуля.

Синтаксис: `$KE,IP,GET`

Ответ на запрос:

#IP,<IpAddress>

Пример:



Получить текущее значение IP адреса модуля:

запрос: \$KE,IP,GET
 ответ: #IP,192.168.0.115



- команда поддерживается последовательным портом

\$KE,MAC,SET

Команда позволяет установить MAC адрес модуля. По умолчанию, MAC адрес модуля равен 00-04-A3-00-00-0B (в десятичном формате 0-4-163-0-0-11). Параметр сохраняется в энергонезависимой памяти. Изменения вступают в силу после перезагрузки модуля (команда \$KE,RST или сброс питания).

Синтаксис: \$KE,MAC,SET,<MacAddress>

Параметры:

MacAddress – MAC адрес в формате X.X.X.X.X.X (в качестве X могут быть использованы числа от 0 до 255). Адреса состоящие из шести нулей или шесть чисел 255 запрещены к использованию.

Ответ на запрос:

#MAC,SET,OK

Пример:



Установить MAC адрес модуля равным 0-4-163-0-0-15:

запрос: \$KE,MAC,SET,0.4.163.0.0.15
 ответ: #MAC,SET,OK



Будьте внимательны при изменении сетевых настроек модуля. Если адрес будет указан некорректно, вы не сможете подключиться к модулю через сетевое соединение. Для сброса/изменения настроек в этом случае следует использовать линию модуля 36 (RST) или последовательный порт.



- команда поддерживается последовательным портом

\$KE,MAC,GET

Возвращает текущий MAC адрес модуля.

Синтаксис: \$KE,MAC,GET

Ответ на запрос:

#MAC,<MacAdress>

Пример:

Получить текущее значение MAC адреса модуля:

запрос: \$KE,MAC,GET
 ответ: #MAC, 0.4.163.0.0.15



- команда поддерживается последовательным портом

\$KE,MSK,SET

Команда позволяет установить маску подсети (Subnet Mask). По умолчанию, маска подсети равна 255.255.255.0. Параметр сохраняется в энергонезависимой памяти. Изменения вступают в силу после перезагрузки модуля (команда \$KE,RST или сброс питания).

Синтаксис: \$KE,MSK,SET,<Mask>**Параметры:**

Mask – Маска подсети в формате X.X.X.X (в качестве X могут быть использованы числа от 0 до 255). Адреса 0.0.0.0 и 255.255.255.255 запрещены к использованию.

Ответ на запрос:

#MSK,SET,OK

Пример:

Установить маску подсети в виде 255.255.255.128:

запрос: \$KE,MSK,SET,255.255.255.128
 ответ: #MSK,SET,OK



Будьте внимательны при изменении сетевых настроек модуля. Если адрес будет указан некорректно, вы не сможете подключиться к модулю через сетевое соединение. Для сброса/изменения настроек в этом случае следует использовать линию модуля 36 (RST) или последовательный порт.



команда поддерживается через последовательный порт

\$KE,MSK,GET

Возвращает текущее значение маски подсети.

Синтаксис: **\$KE,MSK,GET**

Ответ на запрос:

#MSK,<Mask>



- команда поддерживается последовательным портом

\$KE,GTW,SET

Команда позволяет установить шлюз по умолчанию (Default Gateway). Исходно, адрес шлюза равен 192.168.0.1. Параметр сохраняется в энергонезависимой памяти. Изменения вступают в силу после перезагрузки модуля (команда \$KE,RST или сброс питания).

Синтаксис: **\$KE,GTW,SET,<Gateway>**

Параметры:

Gateway – Адрес шлюза в формате X.X.X.X (в качестве X могут быть использованы числа от 0 до 255). Адреса 0.0.0.0 и 255.255.255.255 запрещены к использованию.

Ответ на запрос:

#GTW,SET,OK

Пример:



Установить адрес шлюза виде 192.168.0.12:

запрос: **\$KE,GTW,SET,192.168.0.12**

ответ: **#GTW,SET,OK**



Будьте внимательны при изменении сетевых настроек модуля. Если адрес будет указан некорректно, вы не сможете подключиться к модулю через сетевое соединение. Для сброса/изменения настроек в этом случае следует использовать линию модуля 36 (RST) или последовательный порт.



- команда поддерживается последовательным портом

\$KE,GTW,GET

Возвращает текущее значение адреса шлюза по умолчанию.

Синтаксис: \$KE,GTW,GET

Ответ на запрос:

#GTW,<Gateway>



- команда поддерживается последовательным портом

\$KE,UDT,SET

Синтаксис: \$KE,UDT,SET,<Address>,<Length>,<Data>

Позволяет сохранить произвольные данные размером до 32 байт в энергонезависимой памяти модуля по указанному адресу.

Параметры:

Address – Адрес в памяти, куда следует записать данные. Размер адресной области 256 байт. Поле может принимать значения [0 – 255]

Length – Размеры данных в байтах для записи в память.

Data – данные для записи в память; не более 32 байт

Ответ на запрос:

#UDT,SET,OK

Пример:

Сохранить в энергонезависимой памяти модуля строку 'Hello':

запрос: \$KE,UDT,SET,0,5,Hello

ответ: #UDT,SET,OK

\$KE,UDT,GET

Синтаксис: \$KE,UDT,GET,< Address>,<Length>

Чтение ранее сохраненных пользователем данных из энергонезависимой памяти модуля. Ранее не инициализированная область памяти будет содержать по умолчанию значения 0x00 или 0xFF.

Параметры:

- Address* – Адрес в памяти, с которого следует начинать считывание данных. Размер адресной области 256 байт. Поле может принимать значения [0 – 255]
- Length* – Длина данных для чтения в байтах. Может принимать значения [1-32]

Ответ на запрос:

#UDT,<Size>,<Data>

- Size* – Количество успешно прочтенных байт данных

Пример:

Считать данные энергонезависимой памяти модуля по адресу 0 длиной 20 байт:

запрос: \$KE,UDT,GET,0,20
ответ: #UDT,20,Hello

\$KE,CAT,SET

Команды этой группы позволяют управлять работой системы CAT – работа модуля в автономном режиме с заданной логикой при возникновении событий на входных линиях или по таймеру. Вновь создаваемый объект CAT по умолчанию будет выключен. Параметры объектов CAT сохраняются в энергонезависимой памяти.

Синтаксис 1: \$KE,CAT,<Cat Id>,SET,L,<InLine>,<InEvt>,<OutLine>,<Action>

Позволяет создать событие CAT на входной линии и задать логику срабатывания выходной линии при возникновении события.

Параметры:

- CatId* – Идентификатор события CAT. Система может обрабатывать до 10 событий одновременно. Может принимать значения [1-10].
- InLine* – номер линии ввода/вывода. Должна быть настроена на вход. На этой линии будет детектироваться событие. Может принимать значения [1-22].
- InEvt* – Если значение равно 1: событие произойдет при изменении уровня на входной линии с низкого на высокий. 0 – наоборот, при переходе с высокого на низкий.

- OutLine* – номер линии ввода/вывода. Должна быть настроена на выход. На этой линии будет установлен заданный уровень напряжения при возникновении события. Может принимать значения [1-22].
- Action* – При возникновении события на линии *OutLine* будет автоматически установлен логический ноль если поле равно 0, логическая единица если поле равно 1 и значение будет инвертировано если поле равно 2.

Ответ на запрос:

#CAT,SET,OK

Пример:

Создадим новый объект CAT с идентификатором 2. Событие будет привязано к линии под номером IO_5, переход от лог. 0 к лог.1. В качестве реакции на событие произведем инверсию состояния выходной линии IO_21.

запрос: \$KE,CAT,2,SET,L,5,1,21,2
 ответ: #CAT,SET,OK

Синтаксис 2: \$KE,CAT,<Cat Id>,SET,T,<Timer>,<OutLine>,<Action>

Позволяет создать событие CAT по таймеру и задать логику срабатывания выходной линии при возникновении события.

Параметры:

- CatId* – Идентификатор события CAT. Система может обрабатывать до 10 событий одновременно. Может принимать значения [1-10].
- Timer* – Период срабатывания таймера (генерации события) в секундах от 1 до 15000. При срабатывании таймера будет произведено действие с выходной дискретной линией *OutLine*
- OutLine* – номер линии ввода/вывода. Должна быть настроена на выход. На этой линии будет установлен заданный уровень напряжения при возникновении события. Может принимать значения [1-22].
- Action* – При возникновении события на линии *OutLine* будет автоматически установлен логический ноль если поле равно 0, логическая единица если поле равно 1 и значение будет инвертировано если поле равно 2.

Ответ на запрос:

#CAT,SET,OK

Пример:

Создадим новый объект CAT с идентификатором 6. Событие будет привязано к таймеру с периодом срабатывания 300 секунд. В качестве реакции на событие произведем инверсию состояния выходной линии IO_9.

запрос: \$KE,CAT,6,SET,T,300,9,2
ответ: #CAT,SET,OK

При возникновении события в порт выдается информационное сообщение, имеющие следующий формат:

#ECAT,L,<CatId>,<Counter>

где

- CatId* – Идентификатор события CAT. Система может обрабатывать до 10 событий одновременно. Может принимать значения [1-10].
- Counter* – Значение счетчика событий.

Синтаксис 3: \$KE,CAT,<Cat Id>,GET

Возвращает информацию по событию CAT под индексом *CatId*. Параметр *CatId* может принимать значения [1-10].

Ответ на запрос:

Для CAT элементов с событием на входной линии:

#CAT,<CatId>,<State>,L,<InLine>,<InEvt>,<OutLine>,<Action>

Для CAT элементов с событием по таймеру:

#CAT,<CatId>,<State>,T,<Timer>,<OutLine>,<Action>

- CatId* – Идентификатор события CAT. Система может обрабатывать до 10 событий одновременно. Может принимать значения [1-10].
- State* – Определяет, включено событие или нет. 0 – выключено, 1 – включено.
- InLine* – номер линии ввода/вывода. Должна быть настроена на вход. На этой линии будет детектироваться событие. Может принимать значения [1-22].
- InEvt* – Если значение равно 1: событие произойдет при изменении уровня на входной линии с низкого на высокий. 0 – наоборот, при переходе с высокого на низкий.
- Timer* – Период срабатывания таймера (генерации события) в секундах от 1 до 15000. При срабатывании таймера будет произведено действие с выходной дискретной линией *OutLine*

OutLine – номер линии ввода/вывода. Должна быть настроена на выход. На этой линии будет установлен заданный уровень напряжения при возникновении события. Может принимать значения [1-22].

Action – При возникновении события на линии *OutLine* будет автоматически установлен логический ноль если поле равно 0, логическая единица если поле равно 1 и значение будет инвертировано если поле равно 2.

При возникновении события в порт выдается информационное сообщение, имеющие следующий формат:

#ECAT,T,<CatId>,<Counter>

где

CatId – Идентификатор события CAT. Система может обрабатывать до 10 событий одновременно. Может принимать значения [1-10].

Counter – Значение счетчика событий.

Синтаксис 4: \$KE,CAT,<Cat Id>,<Action>

Команда позволяет включить / выключить / удалить CAT событие под индексом *CatId*. Параметр *CatId* может принимать значения [1-10].

Action – ON – включить, OFF – выключить, DEL - удалить

Ответ на запрос:

#CAT,<Action>,OK

\$KE,CAT,ON/OFF

Команды этой группы позволяют включить или выключить все имеющиеся события CAT.

Синтаксис: \$KE,CAT,<State>

State – 0 – OFF, 1 - ON.

Ответ на запрос:

#CAT,<State>,OK

\$KE,CAC

Команды этой группы позволяют управлять счетчиками событий CAT объектов.

Синтаксис 1: \$KE,CAC,RST

Команда обнуляет значения счетчиков событий для всех объектов CAT.

Синтаксис 2: \$KE,CAC,<CatId>

Команда позволяет запросить значение счетчика для CAT объекта под индексом *CatId*. Параметр *CatId* может принимать значения [1-10].

Ответ на запрос:

#CAC,<CatId>,<Counter>

Counter – Значение счетчика событий.

Синтаксис 3: \$KE,CAC,<CatId>,RST

Команда позволяет обнулить показания счетчика событий для CAT объекта под индексом *CatId*. Параметр *CatId* может принимать значения [1-10].

\$KE,INF

Команда возвращает сводную информацию об имени устройства, версии программного обеспечения и серийном номере.

Синтаксис: \$KE,INF**Ответ на запрос:**

#INF,<DeviceName>,<FW Version>,<SerialNumber>

Параметры:

DeviceName – имя устройства. Установлено в значение “Jerome”.

FW Version – номер версии программного обеспечения модуля

SerialNumber – серийный номер модуля

\$KE,RST

Программный сброс модуля. После подачи команды модуль начинает работу как после отключения питания. Настройки в энергонезависимой памяти не стираются.

Синтаксис: \$KE,RST

\$KE,DEFAULT

Программный сброс модуля с очисткой энергонезависимой памяти. После подачи команды модуль начинает работу как после отключения питания. Настройки в энергонезависимой памяти возвращаются в значение по умолчанию (заводские настройки).

Синтаксис: \$KE,DEFAULT



команда поддерживается последовательным портом



© 2014 **KERNELCHIP** Компоненты и модули для управления, мониторинга и автоматизации

Россия, Москва
<http://www.kernelchip.ru>