

# Ethernet модуль Laurent-2

## TCP/IP команды управления

Версия 1.01  
19 марта 2015

**История документа:**

| Версия | Дата          | Описание                  |
|--------|---------------|---------------------------|
| 1.01   | 19 Марта 2015 | Исходная версия документа |

## Содержание

|                       |    |
|-----------------------|----|
| Версия модуля .....   | 4  |
| Введение .....        | 5  |
| \$KE .....            | 6  |
| \$KE,WR .....         | 6  |
| \$KE,WRA .....        | 7  |
| \$KE,RD .....         | 8  |
| \$KE,RID .....        | 9  |
| \$KE,REL .....        | 10 |
| \$KE,RDR .....        | 11 |
| \$KE,ADC .....        | 11 |
| \$KE,IMPL .....       | 12 |
| \$KE,TMP .....        | 13 |
| \$KE,PWM,SET .....    | 14 |
| \$KE,PWM,GET .....    | 14 |
| \$KE,PFR,SET .....    | 15 |
| \$KE,PFR,GET .....    | 15 |
| \$KE,SPB,SET .....    | 16 |
| \$KE,SPB,GET .....    | 17 |
| \$KE,DAT .....        | 17 |
| \$KE,EVT .....        | 18 |
| \$KE,PSW,SET .....    | 19 |
| \$KE,PSW,NEW .....    | 20 |
| \$KE,SEC,SET .....    | 20 |
| \$KE,SEC,GET .....    | 21 |
| \$KE,SAV,SET .....    | 21 |
| \$KE,SAV,FLS .....    | 22 |
| \$KE,SAV,GET .....    | 23 |
| \$KE,DZG,SET .....    | 23 |
| \$KE,DZG,GET .....    | 23 |
| \$KE,IP,SET .....     | 24 |
| \$KE,IP,GET .....     | 24 |
| \$KE,MAC,SET .....    | 25 |
| \$KE,MAC,GET .....    | 25 |
| \$KE,MSK,SET .....    | 26 |
| \$KE,MSK,GET .....    | 27 |
| \$KE,GTW,SET .....    | 27 |
| \$KE,GTW,GET .....    | 28 |
| \$KE,UDT,SET .....    | 28 |
| \$KE,UDT,GET .....    | 28 |
| \$KE,CAT,SET .....    | 29 |
| \$KE,CAT,ON/OFF ..... | 34 |
| \$KE,CAC .....        | 34 |
| \$KE,INF .....        | 35 |
| \$KE,RST .....        | 35 |
| \$KE,DEFAULT .....    | 35 |

## Версия модуля



Данная редакция документа соответствует модулю *Laurent-2* со следующими характеристиками:

|   |       |        |
|---|-------|--------|
| Версия программного обеспечения<br>(“прошивка”) | ..... | L201   |
| Версия Web-интерфейса                           | ..... | LW2.01 |
| Версия платы:                                   | ..... | Rev.D  |

## Введение

Для управления модулем Laurent-2 предназначен набор команд в текстовом формате, называемых КЕ командами. Для управления модулем с помощью КЕ-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 управления в нем реализована система контроля доступа с помощью пароля. Модуль не выполняет команды управления до тех пор, пока не будет введен корректный пароль.

Любая КЕ команда, отсылаемая модулю, должна начинаться с символов '\$КЕ'. 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).

*\$КЕ,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 Laurent-2 встроен последовательный порт RS-232. 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 портом RS-232 помечены специальным символом:



- команда поддерживается через порт RS-232

## \$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 RS-232

## \$KE,WR

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

С помощью данной команды можно установить высокий ( $Value = 1$ ) или низкий уровень напряжения ( $Value = 0$ ) на выходной линии модуля под номером *OutLine*.

**Параметры:**

- |                |   |  |
|----------------|---|--|
| <i>OutLine</i> | – | номер выходной дискретной линии. Может быть в пределах от 1 до 12 включительно. См. выводы OUT_1 – OUT_12. |
| <i>Value</i>   | – | значение для записи на линию. 1 – высокий уровень напряжения, 0 – низкий уровень напряжения (0 В).         |

**Ответ на запрос:**

#WR,OK – значение успешно установлено.

**Пример:**



Установим высокий уровень напряжения на выходной дискретной линии OUT\_6:

запрос: \$KE,WR,6,1  
ответ: #WR,OK

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

С помощью данной команды можно установить высокий или низкий уровень напряжения на всех выходных линиях 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 выходных линиях модуля:

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 выходных дискретных линиях модуля.

**Синтаксис: \$KE,WRA,<ArrayOfValues>****Параметры:**

*ArrayOfValues* – строка длиной от 1 до 12 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 выходной линии OUT\_1, значение второго символа - на линии OUT\_2 и т.д. Строка может содержать меньшее число символов, чем суммарное число выходных линий, например, строка из 4-х символов позволит установить значение на первых четырех выходных линиях (OUT\_1 – OUT\_4).

**Ответ на запрос:**

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

**Пример 1:**

Установим на всех линиях логическую единицу, кроме линии OUT\_12 для которой установим низкий уровень напряжения:

запрос: \$KE,WRA,11111111110

ответ: #WRA,OK,12

**Пример 2:**

Установим на линиях OUT\_3 и OUT\_12 логическую единицу, остальные линии оставим без изменения:

запрос: \$KE,WRA,xx1xxxxxxxx1

ответ: #WRA,OK,2

**Пример 3:**

Установим на первых 8-ми выходных линиях модуля (OUT\_1 – OUT\_8) 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,<InLine>**

С помощью данной команды можно считать состояние входной дискретной линии модуля под номером *InLine*.

**Параметры:**

*InLine* – номер входной дискретной линии. Может быть в пределах от 1 до 6 включительно. См. выводы IN\_1 – IN\_6.

**Ответ на запрос:**

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



**Пример:**

Считаем состояние входной дискретной линии IN\_2:

запрос: \$KE,RD,2  
ответ: #RD,02,1

**Синтаксис (Вариант 2): \$KE,RD,ALL**

По данной команде модуль произведет последовательный опрос всех входных дискретных линий IN\_1 – IN\_6. Результат выводится в виде сводной строки данных, состоящей из 6 символов (по суммарному числу входных линий). Нумерация позиции символа в строке осуществляется слева на право и соответствует номеру линии.

**Ответ на запрос:**

#RD,<Line1 Value>< Line2 Value>< Line3 Value>.... <Line6 Value>

**Пример:**

Считать информацию со всех входных дискретных линий модуля (IN\_1 – IN\_6):

запрос: \$KE,RD,ALL  
ответ: #RD,110010

Данный пример показывает, что на входных линиях под номером 1, 2, и 5 присутствует высокий логический уровень. На остальных линиях – логический ноль.

**\$KE,RID****Синтаксис (Вариант 1): \$KE,RID,<OutLine>**

С помощью данной команды можно считать состояние выходной дискретной линии под номером *OutLine*.

**Параметры:**

*OutLine* – номер выходной дискретной линии. Может быть в пределах от 1 до 12 включительно. См. выводы OUT\_1 – OUT\_12.

**Ответ на запрос:**

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

**Пример:**



Считаем значение с выходной дискретной 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
```

### Синтаксис (Вариант 2): **\$KE,RID,ALL**

С помощью данной команды можно считать состояние всех выходных дискретных линий за один запрос.

#### Ответ на запрос:

```
#RID,ALL,<Line1 Value><Line2 Value><Line3 Value>.... <Line12 Value>
```

Ответ за запрос содержит информацию по всем 12 выходным линиям в виде сводной строки данных. Нумерация в строке производится слева на право. Первому символу в строке соответствует линия OUT\_1, второму символу линия OUT\_2 и т.д. *Line Value = 0* – на линии установлен низкий логический уровень, *Line Value = 1* – соответственно, высокий логический уровень.

#### Пример:



Считать информацию со всех выходных дискретных линий модуля:

```
запрос: $KE,RID,ALL
ответ:  #RID,ALL,011001000000
```

Данный пример показывает, что на линиях под номером 2, 3 и 6 установлен высокий логический уровень. Соответственно, на остальных – логический ноль.

### **\$KE,REL**

Команда предназначена для управления реле модуля (включение/выключение).

**Синтаксис:** **\$KE,REL,<RelNumber>,<Value>**

#### Параметры:

*RelNumber* – номер реле. Может быть в пределах от 1 до 4 включительно.

*Value* – управляющее значение. 0 – реле выключено, 1 - реле включено

#### Ответ на запрос:

#REL,OK – значение успешно установлено.

#### Пример:



Включим второе реле:

запрос: \$KE,REL,2,1

ответ: #REL,OK



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

### \$KE,RDR

Команда позволяет определить, в каком сейчас состоянии находится реле под номером *ReleNumber* – включено оно или выключено.

**Синтаксис:** \$KE,RDR,<ReleNumber>

#### Параметры:

*ReleNumber* – номер реле. Может быть в пределах от 1 до 4 включительно.

#### Ответ на запрос:

#RID,<ReleNumber>,<State> – запрос состояния реле *ReleNumber* произведено успешно, результат *State*. *State* = 0 – реле выключено, *State* = 1 – соответственно, реле включено.

#### Пример:



Запросим состояние 3-го реле модуля:

запрос: \$KE,RDR,3

ответ: #RDR,3,1

Ответ показывает, что в данный момент 3-е реле включено.

### \$KE,ADC

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

**Синтаксис:** \$KE,ADC,<ChannelNumber>

**Параметры:**

*ChannelNumber* – номер канала АЦП. Может быть в пределах от 1 до 2 включительно. См. выводы модуля ADC\_1 – ADC\_2.

**Ответ на запрос:**

#ADC,<ChannelNumber>,<Value> – на входе канала АЦП модуля *ChannelNumber* установлено напряжение *Value*, В.

**Пример:**



Получить значение АЦП с 1-го канала (вывод ADC\_1):

запрос: \$KE,ADC,1  
ответ: #ADC,3,7.418

В данном примере на входе АЦП ADC\_1 присутствует напряжение 7.418 В.

## **\$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. См. выводы модуля IMPL\_1 – IMPL\_4.

*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 (вывод IMPL\_3):

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

Данный пример показывает что в момент времени 1208 счетчик импульсов IMPL\_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 выдает 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,TMP

Считывание значения датчика температуры в градусах Цельсия.

### Синтаксис: \$KE,TMP

#### Ответ на запрос:

#TMP,<Value> – значение датчика температуры в градусах Цельсия. Если датчик температуры не подключен или не исправен – значение температуры выводится равным -273.

#### Пример:



Получить значение датчика температуры:

запрос: \$KE,TMP  
 ответ: #TMP,23.652

## \$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 (RS-232). 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 порта RS-232 представлена в таблице ниже:

| Значение<br>Value | Скорость порта,<br>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 (RS-232).

**Синтаксис: \$KE,SPB,GET****Ответ на запрос:**

#SPB,&lt;Value&gt;

**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,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, состояние реле, значения всех каналов АЦП, температура 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
         #RD,ALL,100111
         #RID,ALL,110011000111
         #RDR,ALL,1101
         #ADC,1,7.341
         #ADC,2,2.692
         #TMP,28.165
         #IMPL,1,T,2,3612
         #IMPL,2,T,0,0
         #IMPL,3,T,0,0
         #IMPL,4,T,0,27519
         #TIME,615
         .....
```

Информация выводится с частотой в 1 Гц.

### **\$KE,EVT**

**Синтаксис:** **\$KE,EVT,<Sate>**

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.

#### 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 (вывод IN\_4). Новое состояние – логическая единица. При этом в порт будет выдано сообщение в следующем формате:

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

*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 – RS232).

**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 - Laurent):

запрос: \$KE,PSW,SET,Laurent  
ответ: #PSW,SET,OK



команда поддерживается через порт RS-232

## **\$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 – “Laurent”):

запрос: `$KE,PSW,NEW,Laurent,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 использовать джампер сброса, расположенный на лицевой стороне платы модуля Laurent-2 или использовать порт RS232, поддерживающий наиболее важные команды управления.

## **\$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). 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



команда поддерживается через порт RS-232

**\$KE,SEC,GET**

Запрос состояния политики безопасности модуля.

**Синтаксис:** \$KE,SEC,GET

**Ответ на запрос:**

#SEC,<State>

**Параметры:**

*Sate* – 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
- реле

- счетчики импульсов
- ШИМ

Для экономии ресурсов памяти модуля (количество циклов запись-чтение), 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:**

*Sate* – 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

*Включаем 1-ое реле:*

**\$KE,REL,1,1**

*Ожидаем не менее 30 сек и отключаем питание...*

*Включаем питание. Запрашиваем состояние реле:*

**\$KE,RDR,1**

*1-ое реле будет включено:*

#RDR,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:**

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

**\$KE,DZG,SET**

Команда включает / выключает режим программного подавления “дребезга контактов” для входных дискретных линий. Настройка сохраняется в энергонезависимой памяти модуля.

**Синтаксис:** \$KE,DZG,SET,<State>

**Параметры:**

*Sate* – Если он равен *ON* (значение по умолчанию), режим включен. Если параметр равен *OFF* – режим выключен.

**Ответ на запрос:**

#DZG,OK

**\$KE,DZG,GET**

Запрос состояния режима подавления “дребезга контактов”.

**Синтаксис:** \$KE,DZG,GET

**Ответ на запрос:**

#DZG,<State>

**Параметры:**

*Sate* – 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. В этом случае для сброса/изменения параметров следует использовать последовательный порт или джампер сброса.



команда поддерживается через порт RS-232

**\$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 через порт RS-232

## **\$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. В этом случае для сброса/изменения параметров следует использовать последовательный порт или джампер сброса



команда поддерживается через порт RS-232

## **\$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 через порт RS-232

## \$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. В этом случае для сброса/изменения параметров следует использовать последовательный порт или джампер сброса.



команда поддерживается через порт RS-232

**\$KE,MSK,GET**

Возвращает текущее значение маски подсети.

**Синтаксис:** **\$KE,MSK,GET**

**Ответ на запрос:**

#MSK,<Mask>



команда поддерживается через порт RS-232

**\$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. В этом случае для сброса/изменения параметров следует использовать последовательный порт или джампер сброса.



команда поддерживается через порт RS-232

**\$KE,GTW,GET**

Возвращает текущее значение адреса шлюза по умолчанию.

**Синтаксис:** \$KE,GTW,GET

**Ответ на запрос:**

#GTW,<Gateway>



команда поддерживается через порт RS-232

**\$KE,UDT,SET**

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

Позволяет сохранить произвольные данные размером до 32 байт в энергонезависимой памяти модуля (общий доступный объем – 256 байт) по указанному адресу.

**Параметры:**

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

*Length* — Размер 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', разместив ее в самом начале области памяти:

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. Вновь создаваемый объект CAT по умолчанию будет включен. 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 / реле при возникновении события.

#### Параметры:

- CatId* — Идентификатор события CAT. Система может обрабатывать до 20 событий одновременно. Может принимать значения [1-20].
- InLine* — номер входной дискретной линии. Может принимать значения [1-6].
- 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 выходной дискретной линии или номер реле. На этой линии будет установлен заданный уровень напряжения при возникновении события. Может принимать значения [1-12] для выходных линий и [201-204] для реле. Соответственно, значению 201 соответствует RELE\_1, 202 – RELE\_2 и т.д.
- Action* – При возникновении события на линии / реле *OutLine* будет автоматически изменено состояние, в соответствие с таблицей возможных значений, представленной ниже:

| Action |   | Название          | Описание   |
|--------|---|-------------------|--|
| 0      |  | Уровень Лог. 0    | постоянный уровень, логический ноль (нет напряжения на выходе / реле выключено)                                      |
| 1      |  | Уровень Лог. 1    | постоянный уровень, логическая единица (есть напряжение на выходе / реле включено)                                   |
| 2      |  | Инверсный уровень | постоянный уровень, противоположенный текущему   |
| 3      |  | Лог.0 импульс     | импульс (1 сек). Линия устанавливается в лог.0, через 1 сек - в лог. 1 вне зависимости от предшествующего состояния. |
| 4      |  | Лог.1 импульс     | импульс (1 сек). Линия устанавливается в лог.1, через 1 сек - в лог. 0 вне зависимости от предшествующего состояния. |
| 5      |  | Инверсный импульс | импульс (1 сек), с уровнем противоположенным текущему. Через 1 сек уровень вернется в предшествующее состояние.      |

**Ответ на запрос:**

#CAT,SET,OK

**Пример:**

Создадим новый объект CAT с идентификатором 2. Событие будет привязано к линии под номером IN\_5, переход от лог. 0 к лог.1. В качестве реакции на событие произведем инверсию состояния реле RELE\_3:

запрос: \$KE,CAT,2,SET,L,5,1,203,2

ответ: #CAT,SET,OK

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

#ECAT,L,&lt;CatId&gt;,&lt;Counter&gt;

*CatId* – Идентификатор события CAT. Может принимать значения [1-20].

*Counter* – Значение счетчика событий.

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

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

**Параметры:**

- CatId* – Идентификатор события CAT. Система может обрабатывать до 20 событий одновременно. Может принимать значения [1-20].
- Timer* – Период срабатывания таймера (генерации события) в секундах от 1 до 15000. При срабатывании таймера будет произведено действие с выходной дискретной линией / реле под номером *OutLine*
- OutLine* – номер выходной дискретной линии или номер реле. На этой линии будет установлен заданный уровень напряжения при возникновении события. Может принимать значения [1-12] для выходных линий и [201-204] для реле. Соответственно, значению 201 соответствует RELE\_1, 202 – RELE\_2 и т.д.
- Action* – При возникновении события на линии / реле *OutLine* будет автоматически изменено состояние, в соответствии с таблицей возможных значений (см. описание \$KE,CAT синтаксис 1)

**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 OUT\_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,T,<CatId>,<Counter>

*CatId* – Идентификатор события CAT. Может принимать значения [1-20].

*Counter* – Значение счетчика событий.

**Синтаксис 3: \$KE,CAT,<Cat Id>,SET,K,<SensorId>,<Condition>,<Porog>,<Outline>,<Action>**

Позволяет создать событие CAT по превышении (выше / ниже) заданного порога показаний датчика температуры.

**Параметры:**

- CatId* – Идентификатор события CAT. Система может обрабатывать до 20 событий одновременно. Может принимать значения [1-20].
- SensorID* – Номер датчика температуры. Для модуля Laurent-2 может принимать только одно фиксированное значение - 1.
- Condition* – Поле определяет условие, при котором срабатывает система слежения. Может принимать два значения: '>' или '<'. '<' – система сработает, если показания температуры опустились ниже порога. '>' – система сработает, если температура поднялась выше порога.
- Porog* – Пороговое значение температуры, в целых градусах Цельсия. Допустимые значения: -50 - + 150 C°
- OutLine* – номер выходной дискретной линии или номер реле. На этой линии будет установлен заданный уровень напряжения при возникновении события. Может принимать значения [1-12] для выходных линий и [201-204] для реле. Соответственно, значению 201 соответствует RELE\_1, 202 – RELE\_2 и т.д.
- Action* – При возникновении события на линии / реле *OutLine* будет автоматически изменено состояние, в соответствие с таблицей возможных значений (см. описание \$KE,CAT синтаксис 1)

**Ответ на запрос:**

#CAT,SET,OK

**Пример:**

Создадим новый объект CAT с идентификатором 15. Объект будет привязан к событию превышения показаний температуры выше 45 C°. В качестве реакции на событие произведем выключение реле RELE\_2.

запрос: \$KE,CAT,15,SET,K,1,>,45,202,0  
ответ: #CAT,SET,OK

При возникновении события в порт выдается информационное сообщение в формате:

#ECAT,K,<CatId>,<Counter>

- CatId* – Идентификатор события CAT. Может принимать значения [1-20].
- Counter* – Значение счетчика событий.



**Синтаксис 4: \$KE,CAT,<Cat Id>,SET,P,<IP>,<Timer>,<Outline>,<Action>**

Позволяет создать событие CAT, реакция на которое производится при отсутствии ответа на команду PING от удаленного сетевого устройства. Опрос устройства выполняется по таймеру с задаваемым периодом.

**Параметры:**

- CatId* — Идентификатор события CAT. Система может обрабатывать до 20 событий одновременно. Может принимать значения [1-20].
- IP* — IP адрес удаленного устройства, которое следует периодически пинговать
- Timer* — Период вызова PING в минутах от 1 до 250. При срабатывании таймера будет произведен вызов PING процедуры. Если ответ от удаленного устройства отсутствует – производится действие с выходной дискретной линией / реле под номером *OutLine*
- OutLine* — номер выходной дискретной линии или номер реле. На этой линии будет установлен заданный уровень напряжения при возникновении события. Может принимать значения [1-12] для выходных линий и [201-204] для реле. Соответственно, значению 201 соответствует RELE\_1, 202 – RELE\_2 и т.д.
- Action* — При возникновении события на линии / реле *OutLine* будет автоматически изменено состояние, в соответствие с таблицей возможных значений (см. описание \$KE,CAT синтаксис 1)

**Ответ на запрос:**

#CAT,SET,OK

**Пример:**

Создадим новый объект CAT с идентификатором 14. Событие будет привязано к процессу пингования устройства с IP адресом 192.168.1.45 с периодом 60 минут (раз в час). Если от устройства не получен ответ на PING запрос – производится сброс реле RELE\_1.

запрос: \$KE,CAT,14,SET,P,192.168.1.45,60,201,4  
ответ: #CAT,SET,OK

**Синтаксис 5: \$KE,CAT,<Cat Id>,GET**

Возвращает информацию по событию CAT под индексом *CatId*. Параметр *CatId* может принимать значения [1-20].

**Синтаксис 6: \$KE,CAT,<Cat Id>,<Action>**

Команда позволяет включить / выключить / удалить CAT событие под индексом *CatId*. Параметр *CatId* может принимать значения [1-20].

*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 [1-20].

**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 [1-20].

**\$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 “Laurent-2”.

*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 через порт RS-232



© 2015 **KERNELCHIP** Компоненты и модули для управления, мониторинга и автоматизации

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