

# Ethernet модуль Laurent-128

## Ke-команды управления

Версия 1.01  
20 Мая 2019

**История документа:**

<b>Версия</b>	<b>Дата</b>	<b>Описание</b>
1.01	20 Мая 2019	Исходная версия документа

## Содержание

Версия модуля .....	4
Введение .....	5
\$KE .....	6
\$KE,REL .....	7
\$KE,RDR .....	8
\$KE,DEF,REL,SET .....	9
\$KE,DEF,REL,GET .....	10
\$KE,DAT .....	11
\$KE,PSW,SET .....	12
\$KE,PSW,NEW .....	13
\$KE,SEC,SET .....	14
\$KE,SEC,GET .....	15
\$KE,PRT,SET .....	16
\$KE,PRT,GET .....	17
\$KE,IP,SET .....	18
\$KE,IP,GET .....	19
\$KE,MAC,GET .....	20
\$KE,MSK,SET .....	21
\$KE,MSK,GET .....	22
\$KE,GTW,SET .....	23
\$KE,GTW,GET .....	24
\$KE,INF .....	25
\$KE,RST .....	26
\$KE,DEFAULT .....	27

## Версия модуля



Данная редакция документа соответствует модулю Laurent-128 со следующими характеристиками:

Версия программного обеспечения ("прошивка")	.....	LX02
Версия платы:	.....	Rev.B (или старше, если не указано обратное)

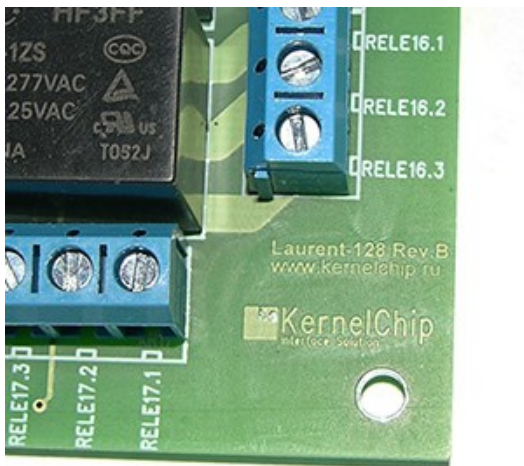


Рис. Обозначение аппаратной ревизии платы Laurent-128 показано на лицевой стороне платы

Серийный номер  
BG78-NJ7A-6ZU2-K892

Версия программного обеспечения  
LX02

Системное время  
26736 с

Рис. Версия "прошивки" показана на лицевой панели Web-интерфейса модуля

## Введение

Для управления модулем Laurent-128 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 TCP порту 2424 (по умолчанию).

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 *Telnet* или *putty*.

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 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*

## **\$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

## \$KE,REL

Команда управления реле.

**Синтаксис:** \$KE,REL,<ReleNumber>,<Value>[,Delay]

### Параметры:

*ReleNumber* – номер реле. Может быть в пределах от 1 до 28 включительно.

*Value* – управляющее значение:  
0 – выключить реле  
1 – включить реле  
2 – инверсия состояния

*Delay* – Необязательный параметр. Задержка в секундах [1-255] по истечении которой реле автоматически будет установлено в исходное состояние

### Ответ на запрос:

#REL,OK – значение успешно установлено.

### Пример 1:

Включим второе реле:

запрос: \$KE,REL,2,1  
ответ: #REL,OK

### Пример 2:

Переведем реле 15 в состояние противоположное текущему на 7 сек после чего реле должно автоматически вернуться в исходное (до подачи команды) состояние:

запрос: \$KE,REL,15,2,7  
ответ: #REL,OK

**\$KE,RDR**

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

**Синтаксис (Вариант 1):** `$KE,RDR,<ReleNumber>`

**Параметры:**

*ReleNumber* – номер реле. Может быть в пределах от 1 до 28 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.

**Синтаксис (Вариант 2):** `$KE,RDR,ALL`

С помощью данной команды можно считать состояние всех реле за один запрос.

**Ответ на запрос:**

`#RDR,ALL,<Rele1 Value><Rele2 Value><Rele3 Value>.... <Rele28 Value>`

Ответ за запрос содержит информацию по всем 28 реле в виде сводной строки данных. Нумерация в строке производится слева на право. Первому символу в строке соответствует 1-ое реле, второму символу 2-ое реле и т.д. *Rele Value* = 0 – реле выключено, *Rele Value* = 1 – включено.

**Пример:**

Запросим состояние всех реле модуля:

```
запрос:  $KE,RDR,ALL
ответ:   #RDR,ALL,01000000000000000000000000000000
```

Ответ показывает, что в данный момент 2-е реле включено, остальные – выключены.



## \$KE,DEF,REL,SET

Команда задает дефолтное состояние реле. Можно настроить реле таким образом что бы оно(они) были включены по умолчанию при подаче питания. Настройка сохраняется в энергонезависимой памяти модуля.

**Синтаксис:** \$KE,DEF,REL,SET,<ArrayOfValues>

### **Параметры:**

*ArrayOfValues* – строка длиной 28 символов. Может содержать символы ‘0’ (выключено), ‘1’ (включено). Нумерация символов в строке производится слева на право. Значение первого символа строки будет установлено на 1-ом реле, значение второго символа - на 2-ом реле и т.д.

### **Ответ на запрос:**

#DEF,REL,SET,OK – значение успешно установлено.

### **Пример:**

Настроим модуль так чтобы 2-ое и 5-ое реле были по умолчанию включены при включении модуля:

запрос: \$KE,DEF,REL,SET,01001000000000000000000000000000  
ответ: #DEF,REL,SET,OK

**\$KE,DEF,REL,GET**

Возвращает ранее заданное дефолтное состояние реле.

**Синтаксис:** \$KE,DEF,REL,GET

**Ответ на запрос:**

#DEF,REL,GET,<ArrayOfValues>

**Параметры:**

*ArrayOfValues* – строка длиной 28 символов. Может содержать символы ‘0’ (выключено), ‘1’ (включено). Нумерация символов в строке производится слева на право. Значение первого символа строки соответствует 1-оум реле, значение второго символа - 2-ому реле и т.д.

**Пример:**

Запросим ранее установленные дефолтные состояния реле:

запрос: \$KE,DEF,REL,GET  
ответ: #DEF,REL,GET,01000000000000000000000000000000

**\$KE,DAT**

Команда включает / выключает выдачу сводной информации по аппаратным ресурсам модуля с частотой 1 Гц. Выводится следующая информация: текущее системное время и состояния реле.

**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,295
        #RDR,ALL,01001000000000000000000000000000
        #TIME,296
        #RDR,ALL,01001000000000000000000000000000
        .....
```

Информация выводится с частотой в 1 Гц.

**\$KE,PSW,SET**

С помощью команды можно ввести пароль доступа к командному интересу модуля (TCP порт 2424 – по умолчанию).

**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,ERR – неверный пароль. Доступ по-прежнему заблокирован

**Пример:**

Введем пароль доступа к модулю (по умолчанию - *Laurent*):

запрос: **\$KE,PSW,SET,Laurent**

ответ: **#PSW,SET,OK**

**\$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. Можно использовать символы 0-9, a-z, A-Z.

**Ответ на запрос:**

- #PSW,NEW,OK – новый пароль успешно установлен
- SPSW,NEW,ERR – 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) – единственным выходом 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.

## **\$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:**

*State* – Если он равен *ON* (значение по умолчанию), то доступ к командному порту TCP 2424, 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`

## **\$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.

## **\$KE,PRT,SET**

Команда позволяет изменять TCP порт для управления модулем (по умолчанию 2424) и web-интерфейса (по умолчанию 80). Данные сохраняются в энергонезависимой памяти. Необходима перезагрузка модуля для вступления изменений в силу (команда \$KE,RST или сброс питания).

**Синтаксис:** \$KE,PRT,<Port Type>,SET,<Value>

### **Параметры:**

*Port Type* – 0 – командный порт, 1 – не используется, 2 – Web

*Value* – Новое значение порта

### **Ответ на запрос:**

#PRT,SET,OK

### **Пример:**

Изменим порт доступа к Web-интерфейсу с 80 на 2000:

запрос: \$KE,PRT,2,SET,2000

ответ: #PRT,SET,OK



## **\$KE,PRT,GET**

Возвращает текущее значение TCP порта управления и Web-интерфейса.

**Синтаксис:** **\$KE,PRT,<Port Type>,GET**

### **Параметры:**

*Port Type* – 0 – командный порт, 1 – не используется, 2 - Web

### **Ответ на запрос:**

#PRT,<Port Type>,<Value>

### **Пример:**

Запросить текущий номер TCP порта для Web-интерфейса:

запрос: **\$KE,PRT,2,GET**  
ответ: **#PRT,2,80**

## **\$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).

### **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 джампер сброса.

## **\$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

## **\$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

## **\$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)

### **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 или джампер сброса.

## **\$KE,MSK,GET**

Возвращает текущее значение маски подсети.

**Синтаксис:** \$KE,MSK,GET

**Ответ на запрос:**

#MSK,<Mask>

**Пример:**

Получить текущее значение маски подсети модуля:

запрос: \$KE,MSK,GET  
ответ: #MSK,255.255.255.0

## \$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)

### **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 или джампер сброса.

## **\$KE,GTW,GET**

Возвращает текущее значение адреса шлюза по умолчанию.

**Синтаксис:** \$KE,GTW,GET

**Ответ на запрос:**

#GTW,<Gateway>

**Пример:**

Получить текущее значение адреса шлюза модуля:

запрос: \$KE,GTW,GET  
ответ: #GTW,192.168.0.1



**\$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-128”.

*FW Version* – номер версии программного обеспечения модуля

*SerialNumber* – серийный номер модуля

**Пример:**

Получить сводную информацию о версии прошивки и серийном номере модуля:

запрос: \$KE,INF

ответ: #INF,Laurent-128,LX02,BG78-NJ7A-6ZU2-K892

## **\$KE,RST**

Программный сброс модуля. Настройки в энергонезависимой памяти не стираются.

**Синтаксис:** \$KE,RST

## **\$KE,DEFAULT**

Программный сброс модуля с очисткой энергонезависимой памяти. Настройки в энергонезависимой памяти возвращаются в значение по умолчанию (заводские настройки).

**Синтаксис: \$KE,DEFAULT**



© 2019 **KERNELCHIP** Компоненты и модули для управления, мониторинга и автоматизации

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