

Ethernet+GSM модуль Laurent-2G

TCP/IP команды управления

Версия 2.01
16 сентября 2016

История документа:

Версия	Дата	Описание
2.01	16 сентября 2016	Добавлена поддержка команды \$KE,DEF
1.01	25 января 2016	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,DEF,REL,SET	15
\$KE,DEF,REL,GET	15
\$KE,DEF,OUT,SET	16
\$KE,DEF,OUT,GET.....	16
\$KE,DAT	17
\$KE,EVT	17
\$KE,PSW,SET	18
\$KE,PSW,NEW.....	19
\$KE,SEC,SET.....	20
\$KE,SEC,GET	20
\$KE,SAV,SET	20
\$KE,SAV,FLS	21
\$KE,SAV,GET	22
\$KE,DZG,SET	22
\$KE,DZG,GET	22
\$KE,IP,SET.....	23
\$KE,IP,GET	23
\$KE,MAC,SET	24
\$KE,MAC,GET	24
\$KE,MSK,SET	25
\$KE,MSK,GET.....	25
\$KE,GTW,SET.....	25
\$KE,GTW,GET	26
\$KE,PIN,SET.....	26
\$KE,PIN,GET.....	27
\$KE,PHN,SET	27
\$KE,PHN,GET	28
\$KE,INIT	28
\$KE,INF.....	29
\$KE,RST.....	29
\$KE,DEFAULT	29

Версия модуля



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

Версия программного обеспечения ("прошивка")	LG07
Версия Web-интерфейса	LW7.01
Версия платы:	Rev.B

Введение

Для управления модулем Laurent-2G 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 или программу KeTerm от KernelChip поддерживающую ОС Windows XP, 7, 8.

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,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:

- OutLine* – номер выходной дискретной линии. Может быть в пределах от 1 до 12 включительно. См. выводы OUT_1 – OUT_12.
- Value* – 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,ОК,12

Пример 2:

Установим на линиях OUT_3 и OUT_12 логическую единицу, остальные линии оставим без изменения:

запрос: \$KE,WRA,xx1xxxxxxxxx1
ответ: #WRA,ОК,2

Пример 3:

Установим на первых 8-ми выходных линиях модуля (OUT_1 – OUT_8) логический ноль:

запрос: \$KE,WRA,00000000
ответ: #WRA,ОК,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. выходы 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,<ReleNumber>,<Value>**

Параметры:

ReleNumber – 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,DEF,REL,SET

Команда задает дефолтное состояние реле при подаче питания на модуль. Настройка сохраняется в энергонезависимой памяти модуля.

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

Параметры:

ArrayOfValues – строка длиной 4 символа. 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 3-ое реле были по умолчанию включены при включении модуля

запрос: \$KE,DEF,REL,SET,0110

ответ: #DEF,REL,SET,OK

\$KE,DEF,REL,GET

Возвращает ранее заданное дефолтное состояние реле при подаче питания на модуль.

Синтаксис: \$KE,DEF,REL,GET

Ответ на запрос:

#DEF,REL,GET,<ArrayOfValues>

Параметры:

ArrayOfValues – строка длиной 4 символа. Может содержать символы ‘0’ (выключено), ‘1’ (включено). Нумерация символов в строке производится слева на право. Значение первого символа строки соответствует 1-ому реле, значение второго символа - 2-ому реле и т.д.

\$KE,DEF,OUT,SET

Команда задает дефолтное состояние выходных дискретных линий при подаче питания на модуль. Настройка сохраняется в энергонезависимой памяти модуля.

Синтаксис: `$KE,DEF,OUT,SET,<ArrayOfValues>`

Параметры:

ArrayOfValues – строка длиной 12 символов. Может содержать символы ‘0’ (выключено), ‘1’ (включено). Нумерация символов в строке производится слева на право. Значение первого символа строки будет установлено на 1-ой выходной линии (OUT_1), значение второго символа - на 2-ой линии и т.д.

Ответ на запрос:

`#DEF,OUT,SET,OK` – значение успешно установлено.

Пример:

Настроим модуль так чтобы 2-ая и 5-ая выходные линии были по умолчанию включены при включении модуля:

запрос: `$KE,DEF,OUT,SET,010010000000`
ответ: `#DEF,OUT,SET,OK`

\$KE,DEF,OUT,GET

Возвращает ранее заданное дефолтное состояние выходных дискретных линий при подаче питания на модуль.

Синтаксис: `$KE,DEF,OUT,GET`

Ответ на запрос:

`#DEF,OUT,GET,<ArrayOfValues>`

Параметры:

ArrayOfValues – строка длиной 12 символов. Может содержать символы ‘0’ (выключено), ‘1’ (включено). Нумерация символов в строке производится слева на право. Значение первого символа строки соответствует 1-ой выходной линии, значение второго символа - 2-ой линии и т.д.

\$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

\$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.

\$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`

\$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,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:

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`

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 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,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 параметров следует использовать аппаратный джампер сброса

\$KE,MAC,GET

Возвращает текущий MAC адрес модуля.

Синтаксис: \$KE,MAC,GET

Ответ на запрос:

#MAC,<MacAddress>

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). 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 параметров следует использовать аппаратный джампер сброса.

\$KE,MSK,GET

Возвращает текущее значение маски подсети.

Синтаксис: `$KE,MSK,GET`

Ответ на запрос:

#MSK,<Mask>

\$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 параметров следует использовать аппаратный джампер сброса.

\$KE,GTW,GET

Возвращает текущее значение адреса шлюза по умолчанию.

Синтаксис: \$KE,GTW,GET

Ответ на запрос:

#GTW,<Gateway>

\$KE,PIN,SET

Команда устанавливает PIN код SIM карты. Модуль в автоматическом режиме производит запуск и инициализацию GSM модема. Для этого ему необходимо знание PIN кода. Данные сохраняются в энергонезависимой памяти.

Синтаксис: \$KE,PIN,SET,<PinCode>

Ответ на запрос:

#PIN,SET,OK

Параметры:

PinCode – PIN код SIM карты. Длина – 4 символа. Допустимы символы [0 - 9]

Пример:



Установим новый PIN код:

запрос: \$KE,PIN,SET,1234

ответ: #PIN,SET,OK

Если SIM карта не предусматривает использование PIN или он отключен – следует указать PIN 0000.

\$KE,PIN,GET

Запрос значения текущего PIN кода SIM карты.

Синтаксис: \$KE,PIN,GET

Ответ на запрос:

#PIN,<PinCode>

Если PIN код еще не был задан, модуль выдает ответ в виде: #PIN,NOTSET

Пример:



Запрос текущего PIN кода:

запрос: \$KE,PIN,GET

ответ: #PIN,1234

\$KE,PHN,SET

Команда устанавливает номер базового телефона. Именно этот номер используется модулем для обратной связи. Данные сохраняются в энергонезависимой памяти.

Синтаксис: \$KE,PHN,SET,<Phone>

Ответ на запрос:

#PHN,SET,OK

Параметры:

Phone – Номер базового телефона. Максимальная длина – 20 символов.
Допустимы символы [0 - 9].

Пример:

Установим номер базового телефона:

запрос: \$KE,PHN,SET,89161366613
ответ: #PHN,SET,OK

\$KE,PHN,GET

Запрос значения текущего номера базового телефона.

Синтаксис: \$KE,PHN,GET

Ответ на запрос:

#PHN,<Phone>

Если номер еще не был задан, модуль выдает ответ в виде: #PHN,NOTSET

Пример:

Запрос значения номера базового телефона:

запрос: \$KE,PHN,GET
ответ: #PHN,89161366613

\$KE,INIT

Команда задает режим работы GSM функционала - разрешает/запрещает запуск GSM модема.
Настройки сохраняются в энергонезависимой памяти.

Синтаксис: \$KE,INIT,<SetupCode>

Ответ на запрос:

#INIT,OK

Параметры:

SetupCode – Числовой параметр, определяющий режим работы GSM. Может принимать значения [0-1]. Допустимые значения:
 0: GSM отключены
 1: GSM включен

Пример:

Разрешим работу GSM модема:

запрос: \$KE,INIT,1
 ответ: #INIT,OK

\$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-2G”.

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



© 2016 **KERNELCHIP** Компоненты и модули для управления, мониторинга и автоматизации

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