Ethernet модуль Laurent-5 / Laurent-5G

Ke-команды управления

Версия 1.11

17 Мая 2021

KernelChip
Interface solutions

История документа:

Версия	Описание
1.11 17 Мая 2021	1. Обновлено описание $KE,OWI,MOD (поддержка DHT-22) 2. Обновлено описание $KE,VAR 3. Удалены команды $KE,VAR,PP и $KE,VAR,MM 4. Добавлено описание команды $KE,IPL,DIR 5. Добавлены новые типы динамических переменных: IL, II и AC
1.10 16 Апреля 2021	6. Добавлено описание команд управления значением счетчика импульсов: - $KE,IPL,VAL,SET - $KE,IPL,VAL,PP - $KE,IPL,VAL,MM 7. Добавлено описание новых Динамических переменных: IC, IS, UV, WG, IB
1.9 22 Марта 2021	1. Новые команды $KE,PWM,PP / $KE,PWM,MM 2. Группа команд управления Ke-скриптами $KE,SCR 3. Исправлена опечатка в команде $KE,PHN – в документе ошибочно использовался некорректный синтаксис *$KE,PNH* 4. Обновлено описание команды $KE,RFD с учетом увеличения объема хранимых меток в базе данных 5. Обновлено описание команды $KE,PHN
1.8 09 Декабря 2020	1. Добавлено описание команд управления ModbusTCP сервера 2. Удалено описание команд группы M2M (более не поддерживается)
1.7 03 Ноября 2020	1. Исправление ошибки с именем динамической переменной *~Dh1~* (показания влажности датчика DHT-11) 2. Дополнено описание команды $KE,URL
1.6 16 Сентября 2020	1. Добавлено описание новых команд: - $KE,SER,BITS - $KE,PWM,DRV - $KE,WGN,TOUT 2. Добавлено описание динамических переменных 3. Обновлено описание ряда команд с учетом возможности применения динамических переменных: - $KE,PUT - $KE,URL

	- $KE,SMS 4. Обновлено описание команды $KE,PUT – добавлена возможность задавать строку как в ACSII, так и в HEX виде
1.5 06 Августа 2020	Добавлено описание новых команд поддержанных в GSM модуле Laurent-5G: - $KE,PIN - $KE,GMD - $KE,GSM,RST - $KE,GSM,STAT - $KE,GSM,ICAL - $KE,GSM,ISMS - $KE,PHN - $KE,RNG - $KE,SMS
1.4 05 Мая 2020	Добавлено описание команд появившихся (изменившихся) в версии “прошивки” L506 / L507: - $KE,SAV - $KE,RFD,GET,NUM - $KE,WGN - $KE,WEB - $KE,ACS
1.3 07 Февраля 2020	Добавлено описание команд появившихся в версии “прошивки” L505: - $KE,NBN - $KE,URL - $KE,CAT,ALL,DEL
1.2 28 Января 2020	Добавлена возможность установки инверсного состояния линии в командах *$KE,WR* и *$KE,WRA*
1.1 06 Ноября 2019	Исходная версия документа

Содержание

Введение 8

Информационные команды 9
$KE 9
$KE,INF 10

Реле 11
$KE,REL 11
$KE,RDR 12

Входные оптоизолированные линии (IN) 13
$KE,RD 13
$KE,IN,TIME,GET 15
$KE,IN,TIME,RST 16

Выходные силовые линии OUT 17
$KE,RID 17
$KE,WR 19
$KE,WRA 20

Дискретные линии общего назначения IO 21
$KE,IOD,SET 21
$KE,IOD,GET 22
$KE,IOW 23
$KE,IOR 24
$KE,IO,TIME,GET 25
$KE,IO,TIME,RST 26

АЦП 27
$KE,ADC,RAW 27
$KE,ADC,VLT 29
$KE,ACF 30

Датчики тока 31
$KE,ACS,MOD 31
$KE,ACS,TYP 32
$KE,ACS,OFS 33

ШИМ 34

$KE,PWM 34
$KE,PWM,PP 36
$KE,PWM,MM 37
$KE,PWM,DRV 38
$KE,PSM 39
$KE,PFR 41

RS-232 43
$KE,SPB,SET 43
$KE,SPB,GET 44
$KE,SER,MOD 45
$KE,SER,RX,GET 47
$KE,SER,TX,GET 48
$KE,SER,RX,RST 49
$KE,SER,TX,RST 50
$KE,SER,RX,TIME,GET 51
$KE,SER,RX,TIME,RST 52
$KE,SER,BITS 53

Счетчик импульсов 55
$KE,IPL,SAV 55
$KE,IPL,RST 58
$KE,IPL,EVT 60
$KE,IPL,ENB 62
$KE,IPL,MEM 64
$KE,IPL,VAL 66
$KE,IPL,DIR 70

1-Wire 72
$KE,OWI,PWR 72
$KE,OWI,MOD 74

1-Wire датчики температуры 76
$KE,TMP,SCAN 76
$KE,TMP,GET,NUM 77
$KE,TMP,ADR 78
$KE,TMP,VAL 79

RFID & iButton 80
$KE,WGN 80
$KE,WGN,TOUT 82
$KE,RFD 84

ModbusTCP 90
$KE,MBS 90
$KE,MBS,PRT 92

Датчик влажности DHT-11 94
$KE,HMD,VAL,GET 94

Система CAT 95
$KE,CAT 95
$KE,CAC 100

TCP клиент 103
$KE,BTCPC 103
$KE,BTCPC,IP 105
$KE,BTCPC,PRT 107

GSM 109
$KE,PIN 109
$KE,GMD 111
$KE,GSM,RST 113
$KE,GSM,STAT 114
$KE,GSM,ICAL 115
$KE,GSM,ISMS 117
$KE,PHN 119
$KE,RNG 122
$KE,SMS 123

Выдача / передача данных 125
$KE,PUT 125
$KE,MSG 127
$KE,URL 129

Ke-скрипты 131
$KE,SCR,RUN 131
$KE,SCR,STOP 132
KE,SCR,DEL 133

Безопастность, права доступа 134
$KE,PSW,SET 134
$KE,PSW,NEW 135
$KE,PSW,GET 136
$KE,PSW,BLK 137
$KE,SEC,SET 138
$KE,SEC,GET 139
$KE,PRT,SET 140
$KE,PRT,GET 141

Сетевые настройки....142
$KE,IP,SET....142
$KE,IP,GET....143
$KE,MAC,GET....144
$KE,MSK,SET....145
$KE,MSK,GET....146
$KE,GTW,SET....147
$KE,GTW,GET....148
$KE,NBN,SET....149
$KE,NBN,GET....150

Дополнительные настройки....151
$KE,DZG....151
$KE,VAR....153
$KE,SAV....155
$KE,WEB....158

Сброс настроек....160
$KE,RST....160
$KE,DEFAULT....160

Ke - сообщения....161
Синтаксис....161
ECAT....163
EIOI....166
RFID....167
IBUT....168
DS18....169
TIME....170
RELE....171
IN....172
IOD....173
IOI....174
IOO....175
OUT....176
ADCR....177
ADCV....178
PWM....179
1WT....180
HMD....181
IPLL....182
ACS....183

Динамические переменные....184
Синтаксис....184
Список переменных....185
Примеры....190

Введение

Помимо управления модулем через встроенный Web-интерфейс, Laurent-5 / 5G поддерживает набор текстовых команд управления называемых Ke-командами (открытый API), которыми можно управлять модулем через различные интерфейсы (TCP сервер / клиент, RS-232 и т.д.). Идеология Ke-команд похожа на AT-команды для GSM модемов.

Например, команда ниже включает 3-ое реле:

```
$KE,REL,3,1
```

а для смены адреса дефолтного шлюза (сетевые настройки модуля) можно воспользоваться командой:

```
$KE,GTW,SET,192.168.0.12
```

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

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.

Информационные команды

Раздел описывает несколько служебных и сервисных Ke-команд, полезных для работы с модулем (проверка связи, запрос серийного номера и версии прошивки).

$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,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-5”.

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-5,L501,BG78-NJ7A-6ZU2-K892

Реле

Команды управления электромагнитынми реле (RELE1 – RELE4) установленными на модуле Laurent-5.

$KE,REL

Команда управления реле.

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

Параметры:

ReleNumber – номер реле. Может быть в пределах от 1 до 4 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 переключено в исходное состояние (противоположенное текущему)

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 3 в состояние противоположенное текущему на 7 сек после чего реле должно автоматически вернуться в исходное (до подачи команды) состояние:

запрос: $KE,REL,3,2,7
ответ: #REL,OK

$KE,RDR

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

Синтаксис (Вариант 1): $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.

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> .... <Rele4 Value>

Ответ за запрос содержит информацию по всем 4 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,0100

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

Входные оптоизолированные линии (IN)

Команды управления и контроля входных оптоизолированных (гальванически развязанных) линий IN1 –IN6 (“сухой” контакт).

$KE,RD

Запрос текущего состояния входных оптоизолированных (гальванически развязанных) входных линий IN1 – IN6.

Синтаксис (Вариант 1): $KE,RD,<InLine>

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

Параметры:

InLine – номер входной дискретной линии. Может быть в пределах от 1 до 6 включительно. См. выводы IN1 – IN6.

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:

Запросим состояние входной оптоизолированной линии IN5:

запрос: $KE,RD,5
ответ: #RD,5,1

Ответ показывает, что в данный момент на вход IN5 подано напряжение (если сигнал).

Синтаксис (Вариант 2): $KE,RD,ALL

По данной команде модуль произведет последовательный опрос всех входных дискретных линий IN1 – IN6. 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>.... <Line6 Value>

Пример:

Запросим состояние всех входной оптоизолированной линий:

запрос: $KE,RD,ALL
ответ: #RD,110010

Данный пример показывает, что на входных линиях IN1, IN2, и IN5 присутствует высокий логический уровень. На остальных линиях – логический ноль.

$KE,IN,TIME,GET

Возвращает время в секундах прошедшее с момента последнего изменения уровня (высокий / низкий) на входной оптоизолированной линии IN1 – IN6.

Синтаксис: $KE,IN,TIME,<InLine>,GET

Ответ на запрос:

#IN,TIME,<InLine>,<Value>

Параметры:

InLine – номер входной дискретной линии. Может быть в пределах от 1 до 6 включительно. См. выводы IN1 – IN6.

Value – Время в секундах прошедшее в с момента последнего изменения уровня сигнала на входной линии *InLine*

$KE,IN,TIME,RST

Сбрасывает (обнуляет) счетчик времени прошедшго с момента последнего изменения уровня (высокий / низкий) на входной оптоизолированной линии IN1 – IN6.

Синтаксис: $KE,IN,TIME,<InLine>,RST

Ответ на запрос:

#IN,TIME,RST,OK

Параметры:

InLine – номер входной дискретной линии. Может быть в пределах от 1 до 6 включительно. См. выводы IN1 – IN6.

Выходные силовые линии OUT

Команды управления и контроля выходных силовых дискретных линий OUT1 – OUT5.

$KE,RID

Запрос текущего состояния выходных силовых линий OUT1 – OUT5.

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 5 включительно. См. выводы OUT1 – OUT5.

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 OUT5:

запрос: $KE,RID,5
ответ: #RID,5,1

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>.... <Line5 Value>

Ответ за запрос содержит информацию по всем 5 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:

Считаем состояние всех выходных силовых дискретных линий модуля:

запрос: $KE,RID,ALL
ответ: #RID,ALL,01100

Данный пример показывает, что на линиях OUT2 и OUT3 установлен высокий логический уровень. Соответственно, на остальных – логический ноль.

$KE,WR

Управление состоянием выходных силовых линий OUT1 – OUT5 (включение / выключние).

Синтаксис: $KE,WR,<OutLine>,<Value>[,Delay]

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 5 включительно. См. выводы OUT1 – OUT5.
Value	–	значение для записи на линию: 0 – низкий уровень напряжения (0 В) 1 – высокий уровень напряжения 2 – инверсия состояния
Delay	–	Необязательный параметр. Задержка в секундах [1-255] по истечении которой линия автоматически будет переведена в исходное состояние (противоположенное вновь установленному)

Ответ на запрос:

#WR,OK – значение успешно установлено.

Пример:

Установим высокий уровень напряжения на выходной дискретной линии OUT_3:

запрос: $KE,WR,3,1
ответ: #WR,OK

$KE,WRA

Команда $KE,WRA позволяет за одно обращение установить произвольную комбинацию высоких или низких уровней напряжения (включить / выключить) на всех выходных силовых линиях OUT1 – OUT5.

Синтаксис: $KE,WRA,<ArrayOfValues>

Параметры:

ArrayOfValues – строка длинной от 1 до 5 символов. Может содержать символы ‘0’ (низкий уровень), ‘1’ (высокий уровень), ‘2’ (инверсный 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_2 для которой установим низкий уровень напряжения:

запрос: $KE,WRA,10111
ответ: #WRA,OK,5

Пример 2:

Установим на линиях OUT_2 и OUT_3 логическую единицу, остальные линии оставим без изменения:

запрос: $KE,WRA,x11xx
ответ: #WRA,OK,2

Пример 3:

Установим на первых 3-х выходных линиях модуля (OUT_1 – OUT_3) логический ноль:

запрос: $KE,WRA,000
ответ: #WRA,OK,3

Дискретные линии общего назначения IO

Команды управления и контроля дискретных линий общего назначения IO1 – IO8 настроенных на вход.

$KE,IOD,SET

Команда позволяет настроить дискретную линию ввода/вывода общего назначения IO 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: $KE,IOD,SET,<LineNumber>,<IoDirection>

Параметры:

LineNumber – номер линии ввода/вывода. Может быть в пределах от 1 до 8 включительно.

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

Ответ на запрос:

#IOD,SET,OK

Пример:

Установить дискретную линию ввода/вывода под номером 5 (IO5) в качестве выходной:

запрос: $KE,IOD,SET,5,0
ответ: #IOD,SET,OK

Пример:

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

запрос: $KE,IOD,SET,3,1
ответ: #IOD,SET,OK

$KE,IOD,GET

Возвращает текущее направление (вход или выход) для указанной дискретной линии ввода/вывода общего назначения IO.

Синтаксис: $KE,IOD,GET,<LineNumber>

Параметры:

LineNumber – номер линии ввода/вывода. Может быть в пределах от 1 до 8 включительно.

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

Ответ на запрос:

#IOD,<LineNumber>,<IoDirection>

Пример:

Получить значение направления линии общего назначения IO5:

запрос: $KE,IOD,GET,5
ответ: #IOD,5,1

Пример показывает, что линия настроена на вход.

$KE,IOW

Команда позволяет установить высокий (включена) или низкий (выключена) логический уровень напряжения для дискретной линии ввода/вывода общего назначения IO настроенной на выход.

Синтаксис: $KE,IOW,< LineNumber >,<Value>[,Delay]

Параметры:

LineNumber – номер IO линии настроенной на выход. Может быть в пределах от 1 до 8 включительно. См. выводы IO1 – IO8. Если в текущий момент времени линии настроена на вход – команда вернет сообщение об ошибке.

Value – значение для записи на линию:
- 0 – выключить / низкий уровень напряжения (0 В)
- 1 – включить / высокий уровень напряжения
- 2 – инверсия состояния

Delay – Необязательный параметр. Задержка в секундах [1-255] по истечении которой линия автоматически будет переведена в исходное состояние (противоположенное вновь установленному)

Ответ на запрос:

#IOW,OK – значение успешно установлено

Пример:

Установим высокий уровень напряжения на линии IO3 настроенной на выход:

запрос: $KE,IOW,3,1
ответ: #IOW,OK

$KE,IOR

С помощью данной команды можно считать информацию с дискретной линии ввода/вывода общего назначения. 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. Если линия настроена на выход, то команда вернет последнее значение, которое было установлено для этой линии с помощью команд $KE,WR или $KE,WRA.

Синтаксис: $KE,IOR,<LineNumber>

Параметры:

LineNumber – номер IO линии. Может быть в пределах от 1 до 8 включительно. См. выводы IO1 – IO8.

Ответ на запрос:

#IOR,<LineNumber>,<Value> – состояние линии *LineNumber* равно *Value.*

Value = 0 – Если линия настроена на выход – на ней установлен низкий уровень напряжения (линия выключена). Если линия настроена на вход – на входе линии (“снаружи” модуля) присутствует логический ноль (сигнал отсутствует).

Value = 1 – Если линия настроена на выход – на ней установлен высокий уровень напряжения (линия включена). Если линия настроена на вход – на входе линии (“снаружи” модуля) присутствует сигнал (логическая 1).

Пример:

Установим направление линии IO2 “на выход”, установим на ней высокий логический уровень (включим ее) и запросим текущее состояние:

запрос: $KE,IOR,2
ответ: #IOR,2,1

$KE,IO,TIME,GET

Возвращает время в секундах прошедшее с момента последнего изменения уровня (высокий / низкий) на дискретной линии ввода/вывода общего назначения IO настроенной на вход.

Синтаксис: $KE,IO,TIME,<IoLine>,GET

Ответ на запрос:

#IO,TIME,<IoLine>,<Value>

Параметры:

IoLine – номер IO линии. Может быть в пределах от 1 до 8 включительно. См. выводы IO1 – IO8.

Value – Время в секундах прошедшее с момента последнего изменения уровня сигнала на линии IO под номером *IoLine* настроенной на вход

$KE,IO,TIME,RST

Сбрасывает (обнуляет) счетчик времени прошедшго с момента последнего изменения уровня (высокий / низкий) на дискретной линии ввода/вывода общего назначения IO настроенной на вход.

Синтаксис: $KE,IO,TIME,<IoLine>,RST

Ответ на запрос:

#IO,TIME,RST,OK

Параметры:

IoLine – номер IO линии. Может быть в пределах от 1 до 8 включительно. См. выводы IO1 – IO8.

АЦП

Команды управления и контроля АЦП модуля.

$KE,ADC,RAW

Считывание результата каналов 10-битного АЦП модуля в виде “сырого” цифрового кода в диапазоне значений [0 - 1023].

Синтаксис 1: $KE,ADC,RAW,ALL

Ответ на запрос:

#ADC,RAW,ALL,<Value_1>,< Value_2>,….,< Value_5> – цифровой код АЦП для канала 1 (ADC1) равен Value_1 и т.д. для всех 5-ти каналов АЦП модуля.

Пример:

Запросить текущие цифровые коды АЦП для весх каналов за одно обращение:

запрос: $KE,ADC,RAW,ALL
ответ: #ADC,RAW,ALL,0,0,685,0,893

В данном примере цифровой код преобразования для каналов ADC_1, 2, 4 составляет ноль, для остальных каналов – соответствующие не нулевые значения.

Синтаксис 2: $KE,ADC,RAW, <ChannelNumber>

Параметры:

ChannelNumber – номер канала АЦП. Может быть в пределах от 1 до 5 включительно. См. выводы модуля ADC_1 – ADC_5.

Ответ на запрос:

#ADC,RAW,<ChannelNumber>,<Value> – цифровой код преобразования для канала АЦП модуля *ChannelNumber* равен *Value*

Пример:

Получить значение цифрового кода АЦП с 1-го канала (вывод ADC_1):

запрос: $KE,ADC,RAW,1
ответ: #ADC,RAW,1,300

В данном примере цифровой код преобразования АЦП для канала ADC_1 равен 300.

$KE,ADC,VLT

Считывание результата преобразования АЦП в виде значения напряжения (в Вольтах) пересчитанного из “сырого” цифрового кода с учетом значения напряжения источника опорного напряжения, аппартных и пользовательских коэффициентов делителя напряжения.

Синтаксис 1: $KE,ADC,VLT,ALL

Ответ на запрос:

#ADC,RAW,VLT,<Value_1>,< Value_2>,….,< Value_5> – на входе канала 1 АЦП модуля установлено напряжение Value_1 (в Вольтах) и т.д. для всех 5-ти каналов АЦП модуля.

Пример:

Запросить текущие оценки напряжения сигнала на входах АЦП для весх каналов за одно обращение:

запрос: $KE,ADC,FLT,ALL
ответ: #ADC,FLT,ALL,0.000,0.000,2.456,0.000,0.000

В данном примере на входе канала АЦП ADC_3 присутствует сигнал с напряжением 2.456 В, на остальных каналах напряжение равно 0 В.

Синтаксис 2: $KE,ADC,VLT, <ChannelNumber>

Параметры:

ChannelNumber – номер канала АЦП. Может быть в пределах от 1 до 5 включительно. См. выводы модуля ADC_1 – ADC_5.

Ответ на запрос:

#ADC,VLT,<ChannelNumber>,<Value> – напряжение сигнала для канала АЦП модуля *ChannelNumber* равно *Value* Вольт.

Пример:

Получить значение напряжения для АЦП с 1-го канала (вывод ADC_1):

запрос: $KE,ADC,FLT,1
ответ: #ADC,FLT,1,1.721

В данном примере напряжение на канале ADC_1 равно 1.721 Вольта.

$KE,ACF

Установка / чтение программного пользовательского коэффициента на который могут быть скорректировнны (умножены) показания АЦП при пересчете цифрового кода в напряжение в Вольтах.

Синтаксис 1: $KE,ACF,< ChannelNumber >,SET,<Value>

Параметры:

ChannelNumber – номер канала АЦП. Может быть в пределах от 1 до 5 включительно. См. выводы модуля ADC_1 – ADC_5.

Value – Значение поправочного коэффициента, безразмерная величина

Ответ на запрос:

#ACF,SET,OK

Пример:

Предположим, что на входе АЦП ADC_2 установлен внешний аппаратный резистивный делить напряжения, который делит входное напряжение на 2.35. Необходимо внести эту поправку в программу модуля и получать оценки напряжения АЦП с учетом поправки:

запрос: $KE,ACF,2,SET,2.35
ответ: #ACF,SET,OK

Синтаксис 2: $KE,ACF,< ChannelNumber >,GET

Параметры:

ChannelNumber – номер канала АЦП. Может быть в пределах от 1 до 5 включительно. См. выводы модуля ADC_1 – ADC_5.

Ответ на запрос:

#ACF,<ChannelNumber>,<Value> – текущее значение поправочного коэффициента для канала АЦП под номером *ChannelNumber* равно *Value*

Пример:

Запрос текущего поправочоного коэффициента для канала АЦП ADC_2:

запрос: $KE,ACF,2,GET
ответ: #ACF,2,2.35

Датчики тока

Команды управления и контроля датчиков тока. Датчики тока подключаются к АЦП модуля (каналы ADC_2 – ADC_5).

$KE,ACS,MOD

Указание типа тока (перемнный / постоянный) для измерения указанным датчиком тока.

Синтаксис 1: $KE,ACS,< ADC_ID >,MOD,SET,<Value>

Параметры:

ID – номер канала АЦП к которому подключен датчик тока. Может быть в пределах от 2 до 5 включительно. См. выводы модуля ADC_2 – ADC_5.

Value – Значение для установки. Возможные значения:

```
OFF – датчик тока не подключен
 AC – переменный ток (50 Гц)
 DC – постоянный ток
```

Ответ на запрос:

#ACS,MOD,SET,OK

Пример:

Предположим, что к входу АЦП ADC_2 подключен датчик тока которым планируется измерять переменный ток. Сообщим об этом модулю:

запрос: $KE,ACS,2,MOD,SET,AC
ответ: #ACS,MOD,SET,OK

Синтаксис 2: $KE,ACS,< ADC_ID >,MOD,GET

Возвращает текущее состояние класса измеряемого тока (постоянный / переменный) для датчика тока подключенного к АЦП каналу ADC_ID.

Ответ на запрос:

#ACS,MOD,< ADC_ID >,< Value >

$KE,ACS,TYP

Указание класса (модели) датчика тока.

Синтаксис 1: $KE,ACS,< ADC_ID >,TYP,SET,<Value>

Параметры:

ID	–	номер канала АЦП к которому подключен датчик тока. Может быть в пределах от 2 до 5 включительно. См. выводы модуля ADC_2 – ADC_5.

Value	–	Значение для установки. Возможные значения:
		`NOT_SET – датчик тока не подключен`
		`KACS-05 – датчик тока KernelChip KACS-05`
		`KACS-20 – датчик тока KernelChip KACS-20`
		`KACS-30 – датчик тока KernelChip KACS-30`

Ответ на запрос:

#ACS,TYP,SET,OK

Пример:

Предположим, что к входу АЦП ADC_3 подключен датчик тока KernelChip KACS-20. Сообщим об этом модулю:

запрос: $KE,ACS,3,TYP,SET,KACS-20
ответ: #ACS,TYP,SET,OK

Синтаксис 2: $KE,ACS,< ADC_ID >,TYP,GET

Возвращает текущее состояние настройки типа (модели) датчика тока.

Ответ на запрос:

#ACS,TYP,< ADC_ID >,< Value >

$KE,ACS,OFS

Величина “смещение нуля” датчика тока. В виду разброса параметров, особенностей окружаещей среды или особенностей конструкции датчика он может выдавать несколько смещенные измерения (даже в состоянии “покоя” когда через него ток не протекает). Данная настройка позволяет индивидуально откалибровать данчик и сохранить в энергонезависимой памяти поправочное смещение которое далее будет применяться для корректировки измерений.

Синтаксис 1: $KE,ACS,< ADC_ID >,OFS,SET,<Value>

Параметры:

ID	–	номер канала АЦП к которому подключен датчик тока. Может быть в пределах от 2 до 5 включительно. См. выводы модуля ADC_2 – ADC_5.
Value	–	Смещение в мА. Допустимые значения: 0 – 65534 мА.

Ответ на запрос:

#ACS,OFS,SET,OK

Пример:

Предположим, что к входу АЦП ADC_3 подключен датчик тока который в состоянии покоя (ток не протекает) показывет “ложные” измерения о токе в 70 мА. Сообщим об этом модулю:

запрос: $KE,ACS,3,OFS,SET,70
ответ: #ACS,OFS,SET,OK

Синтаксис 2: $KE,ACS,< ADC_ID >,OFS,GET

Возвращает текущее значение смещения “нуля” измерений датчика тока.

Ответ на запрос:

#ACS,OFS,< ADC_ID >,< Value >

ШИМ

Команды управления и контроля каналов ШИМ.

$KE,PWM

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

Синтаксис 1: **$KE,PWM,<LineNumber>,SET,<PowerValue>**

Параметры:

LineNumber	–	Номер ШИМ канала, может быть в пределах 1- 4 (совмещенные с функцией ШИМ выводы OUT1 – OUT4)
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 на выводе PWM_2 (совмещен с линией OUT2) – 2-ой канал ШИМ:

запрос: $KE,PWM,2,SET,60
ответ: #PWM,SET,OK

Синтаксис 2: $KE,PWM,<LineNumber>,GET

Возвращает текущее значение мощности ШИМ сигнала.

Ответ на запрос:

#PWM,<LineNumber>,<PowerValue>

Параметры:

LineNumber	–	Номер ШИМ канала, может быть в пределах 1- 4 (совмещенные с функцией ШИМ выводы OUT1 – OUT4)
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.

Пример:

Запросим текущйи уровень мощности ШИМ для 2-го канала:

запрос: $KE,PWM,2,GET
ответ: #PWM,2,60

$KE,PWM,PP

Управление ШИМ выходом модуля – увеличение текущего уровня (0-100%, см. команду $KE,PWM) на единицу (+1). При достижении максимального уровня (100%) команда не будет иметь эффекта.

Синтаксис: $KE,PWM,<LineNumber>,PP

Параметры:

LineNumber – Номер ШИМ канала, может быть в пределах 1- 4 (совмещенные с функцией ШИМ выводы OUT1 – OUT4)

Ответ на запрос:

#PWM,PP,OK

Пример:

Увеличить текущий уровень мощности ШИМ сигнала для 2-го канала на единицу:

запрос: $KE,PWM,2,PP
ответ: #PWM,PP,OK

$KE,PWM,MM

Управление ШИМ выходом модуля – уменьшение текущего уровня (0-100%, см. команду $KE,PWM) на единицу (-1). При достижении минимального уровня (0%) команда не будет иметь эффекта.

Синтаксис: $KE,PWM,<LineNumber>,MM

Параметры:

LineNumber – Номер ШИМ канала, может быть в пределах 1- 4 (совмещенные с функцией ШИМ выводы OUT1 – OUT4)

Ответ на запрос:

#PWM,MM,OK

Пример:

Уменьшить текущий уровень мощности ШИМ сигнала для 2-го канала на единицу:

запрос: $KE,PWM,2,MM
ответ: #PWM,PP,OK

$KE,PWM,DRV

Плавное автоматическое управление ШИМ выходом модуля. Команда задает начальное значение ШИМ, конечное и длительность временного интервала в сеукндах за которое должно произойти задданое изменение уровня.

Синтаксис: $KE,PWM,<LineNumber>,DRV,<Start>,<Stop>,<Time>

Параметры:

LineNumber	–	Номер ШИМ канала, может быть в пределах 1- 4 (совмещенные с функцией ШИМ выводы OUT1 – OUT4)
Start	–	Начальное значение уровня ШИМ (0-100)
PowerValue	–	Конечное значение уровня ШИМ (0-100)
Time	–	Время в секундах (0 - 255) за которое должно произойти плавное изменение уровня от значения *Start* до значения *Stop*

Ответ на запрос:

#PWM,DRV,OK

Пример:

Установить 100% уровень мощности ШИМ сигнала на выводе PWM_2 (совмещен с линией OUT2) – 2-ой канал ШИМ и за 20 секунд плавно уменьшить его до нуля:

запрос: $KE,PWM,2,DRV,100,0,20
ответ: #PWM,DRV,OK

$KE,PSM

Включение / выключении функции ШИМ для совмещенных функционально линий OUT1 – OUT4.

Синтаксис 1: **$KE,PSM,<LineNumber>,SET,<Value>**

Параметры:

LineNumber	–	Номер ШИМ канала, может быть в пределах 1- 4 (совмещенные с функцией ШИМ выводы OUT1 – OUT4)
Value	–	0 – выключить функцию ШИМ для данного канала; 1 - включить

Ответ на запрос:

#PSM,SET,OK

Пример:

Включить функцию ШИМ для 3-го канала:

запрос: $KE,PSM,3,SET,1
ответ: #PSM,SET,OK

Синтаксис 2: $KE,PSM,<LineNumber>,GET

Возвращает текущее состояние канала ШИМ (функция включена / выключена).

Ответ на запрос:

#PSM,<LineNumber>,<Value>

Параметры:

LineNumber	–	Номер ШИМ канала, может быть в предела 1- 4 (совмещенные с функцией ШИМ выводы OUT1 – OUT4)
Value	–	0 – функция ШИМ для данного канала выключена; 1 - включена

Пример:

Запросим текущее состояние функции ШИМ для 3-го канала:

запрос: $KE,PSM,3,GET
ответ: #PSM,3,1

$KE,PFR

Команда позволяет изменять частоту ШИМ сигнала. Настройка применяется ко всем каналам ШИМ. Установленное значение сохраняется в энергонезависимой памяти.

Синтаксис 1: **$KE,PFR,SET,<Value>**

Параметры:

Value – Частота ШИМ в Герцах. Допустимые значения 1 – 10 000 Гц

Ответ на запрос:

#PFR,SET,OK

Пример:

Установим частоту генерации ШИМ импульсов равной 5 Гц:

запрос: $KE,PFR,SET,5
ответ: #PFR,SET,OK

Синтаксис 2: **$KE,PFR,GET**

Возвращает текущую частоту генерации ШИМ.

Ответ на запрос:

#PFR,<Value>

Параметры:

Value – Частота ШИМ в Герцах.

Пример:

Запросим текущую частоту ШИМ:

запрос: $KE,PFR,GET
ответ: #PFR,400

RS-232

Команды управления и контроля работой последовательногно порта RS-232.

$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, определяющая скорость последовательного порта. Может принимать значения от 1 до 7 включительно. Связь параметра *Value* и скорости порта RS-232 представлена в таблице ниже:

Значение Value	Скорость порта, бит/с
0	1200
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,<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,SER,MOD

Команда задает режим работы порта RS-232: командный порт или прозрачный интерфейс TCP-2-COM.

Синтаксис 1: **$KE,SER,MOD,SET,<Value>**

Параметры:

Value – 0 – установить порт в режим работы TCP-2-COM
1 – установить порт в командный режим (прием / обработка Ke-команд)

Ответ на запрос:

#SER,MOD,SET,OK

Пример:

Установить порт RS-232 в командный режим:

запрос: $KE,SER,MOD,SET,1
ответ: #SER,MOD,SET,OK

Синтаксис 2: **$KE,SER,MOD,GET**

Возвращает текущий режим работы порта RS-232.

Ответ на запрос:

#SER,MOD,<Value>

Параметры:

Value – 0 – порт установлен в режим работы TCP-2-COM
1 – порт работает в командном режиме (прием / обработка Ke-команд)

Пример:

Запросить текущий режим работы порта RS-232:

запрос: $KE,SER,MOD,GET
ответ: #SER,MOD,0

В текущий момент времени порт работает в режиме TCP-2-COM.

$KE,SER,RX,GET

Возвращает значения счетчика принятых байт входного (Rx) буфера данных порта RS-232..

Синтаксис: $KE,SER,RX,GET

Ответ на запрос:

#SER,RX,<RX_Value>

Параметры:

RX_Value – Число байт данных принятых модулем по интерфейсу RS-232

Пример:

Запросить текущий входящий трафик по порту RS-232:

запрос: $KE,SER,RX,GET
ответ: #SER,RX,567

На текущий момент времени модуль принял 567 байт по порту RS-232.

$KE,SER,TX,GET

Возвращает значения счетчика отправленных байт выходного (Tx) буфера данных порта RS-232.

Синтаксис: $KE,SER,TX,GET

Ответ на запрос:

#SER,TX,<TX_Value>

Параметры:

TX_Value – Число байт данных отправленных модулем по интерфейсу RS-232

Пример:

Запросить текущий выходящий трафик по порту RS-232:

запрос: $KE,SER,TX,GET
ответ: #SER,TX,9

На текущий момент времени модуль отправил 9 байт данных по порту RS-232.

$KE,SER,RX,RST

Сбрасывает счетчик входящего (Rx) трафика для порта RS-232.

Синтаксис: $KE,SER,RX,RST

Ответ на запрос:

#SER,RX,RST,OK

Пример:

Сбросить показания счетчиков байт выходящего трафика по порту RS-232:

запрос: $KE,SER,RX,RST
ответ: #SER,RX,RST,OK

$KE,SER,TX,RST

Сбрасывает счетчик исходящего (Tx) трафика для порта RS-232.

Синтаксис: $KE,SER,TX,RST

Ответ на запрос:

#SER,TX,RST,OK

Пример:

Сбросить показания счетчиков байт исходящего трафика по порту RS-232:

запрос: $KE,SER,TX,RST
ответ: #SER,TX,RST,OK

$KE,SER,RX,TIME,GET

Возвращает время в секундах с момента последнего приема байта данных по порту RS-232. Команда может быть полезна для управления и настройки работы системы CAT для события “R” (отсутствие активности по порту RS-232).

Синтаксис: $KE,SER,RX,TIME,GET

Ответ на запрос:

#SER,RX,TIME,<RX_Last>

Параметры:

RX_Last – Время в секундах прошедшее с момента последнего приема байта данных по порту RS-232

$KE,SER,RX,TIME,RST

Сбрасывает (обнуляет) счечтик времени прошедшего с момента последнего приема байта данных по порту RS-232. Команда может быть полезна для управления и настройки работы системы CAT для события “R” (отсутствие активности по порту RS-232).

Синтаксис: $KE,SER,RX,TIME,RST

Ответ на запрос:

#SER,RX,TIME,RST,OK

$KE,SER,BITS

Команда позволяет изменять настройки данных последовательно порта (RS-232) а именно число бит данных, стоповых бит и проверку на четность. Настройки сохраняется в энергонезависимой памяти модуля.

Синтаксис 1: **$KE,SER,BITS,SET,<dbits>,<sbits>,<parity>**

Параметры:

dbits – Число бит данных. Возможные значения: 8 и 9 бит. Значение по умолчанию – 8 бит.

sbits – Число стоповых бит. Возможные значения: 1 и 2 бит. Значение по умолчанию – 1 бит

parity – Проверка на четность. Возможные значения:
‘n’ – без проверки (none)
‘e’ – четный (even)
‘o’ – нечетный (odd)

Значение по умолчанию – ‘n’ (без проверки). Проверка на четность и нечетность возможна только для режима с 8 битами данных. В случае использования 9 бит данных – проверка на четность не применяется.

Ответ на запрос:

#SER,BITS,SET,OK

Пример:

Установим настройки линии данных последовательно порта, так что бы использовались 8 бит данных, 1 стоповый бит и проверка на четность:

запрос: $KE,SER,BITS,SET,8,1,e
ответ: #SER,BITS,SET,OK

Синтаксис 2: $KE,SER,BITS,GET

Запрос текущего значения настроек даннах последовательного порта модуля (RS-232).

Ответ на запрос:

#SER,BITS,<dbits>,<sbits>, <parity>

Параметры:

См. описание команды $KE,SER,BITS,SET

Пример:

Запросим текущее значение настроек данных последовательного порта модуля:

запрос: $KE,SER,BITS,GET
ответ: #SER,BITS,8,1,n

Пример показывает, что используется режим с 8 битами данных, одним стоповым битом без проверки на четность.

Счетчик импульсов

Команды управления и контроля счетчиков импульсов для входных оптоизолированных линий и линий общего назначения настроенных на вход.

$KE,IPL,SAV

Команда управляет процессом сохранения значений счетчиков импульсов в энергонезависимую память модуля.

Синтаксис 1: **$KE,IPL,SAV,SET,<Value>**

Параметры:

Value	–	Частота сохранения значений счетчика импульсов в энергонезависимую память в минутах. Допустимые значения [0 - 255]. Значение 0 запрещает сохранение значений счетчиков в память.

Ответ на запрос:

#IPL,SAV,SET,OK

Пример:

Установить частоту сохранения изменений в показаниях счетчиков импульсов в значение раз в 10 минут:

запрос: $KE,IPL,SAV,SET,10
ответ: #IPL,SAV,SET,OK

Синтаксис 2: $KE,IPL,SAV,GET

Команда возвращает текущее значение настройки периода сохранения изменений значений счетчиков импульсов в энергонезависимую память.

Ответ на запрос:

#IPL,SAV,<Value>

Параметры:

Value	–	Частота сохранения значений счетчика импульсов в энергонезависимую память в минутах. Допустимые значения [0 - 255]. Значение 0 запрещает сохранение значений счетчиков в память.

Пример:

Считать текущее значение периода сохранения:

запрос: $KE,IPL,SAV,GET
ответ: #IPL,SAV,10

Изменения в показаниях (значениях) счетчиков импульсов будут сохраняться в энергонезависимую память раз в 10 минут.

Синтаксис 3: **$KE,IPL,SAV,NOW**

Принудительное безотлагательное единоразовое сохранение текущих значений счетчиков импульсов в энергонезависимую память.

Ответ на запрос:

#IPL,SAV,NOW,OK

Пример:

Не дожидаясь следующего цикла сохранения данных, произвести запись значений счетчиков в память:

запрос: $KE,IPL,SAV,NOW
ответ: #IPL,SAV,NOW,OK

$KE,IPL,RST

Сброс показаний счетчиков вимпульсов.

Синтаксис 1: **$KE,IPL,RST,<Type>,ALL**

Параметры:

Type	–	Тип входной линии с которой ассоциирован данный счетчик(и): ‘L’ – оптоизолированные входные линии IN1 – IN6 ‘I’ – дискретные линии общего назначения IO1 – IO8 настроенные на вход

Ответ на запрос:

#IPL,RST,ALL,OK

Пример:

Обнулить все показания всех счетчиков импульсов для дискретных линий общего назначения IO1 – IO8 настроенных на вход:

запрос: $KE,IPL,RST,I,ALL
ответ: #IPL,RST,ALL,OK

Синтаксис 2: **$KE,IPL,RST,<Type>,<Line>**

Параметры:

Type	–	Тип входной линии с которой ассоциирован данный счетчик(и): ‘L’ – оптоизолированные входные линии IN1 – IN6 ‘I’ – дискретные линии общего назначения IO1 – IO8 настроенные на вход
Line	–	Номер счетчика для данного типа входной линии у которого нужно обнулить показания

Ответ на запрос:

#IPL,RST,OK

Пример:

Обнулить показания счетчика импульсов на входной оптоизолированной линии IN6:

запрос: $KE,IPL,RST,L,6
ответ: #IPL,RST,OK

$KE,IPL,EVT

Управление настройкой типа события для счетчика импульсов (срабатывание счечтика по переднему, заднему или обоим фронтам импульса на входе).

Синтаксис 1: **$KE,IPL,EVT,<Type>,<Line>,SET,<Value>**

Установка настройки типа события для указанного счетчика импульсов.

Параметры:

Type	–	Тип входной линии с которой ассоциирован данный счетчик(и): 'L' – оптоизолированные входные линии IN1 – IN6 'I' – дискретные линии общего назначения IO1 – IO8 настроенные на вход
Line	–	Номер счетчика для данного типа входной линии
Value	–	Счетчик будет реагировать на: 0 – переход уровня сигнала на входе с низкого на высокий 1 – с высокого на низкий 2 – оба варианта

Ответ на запрос:

#IPL,EVT,SET,OK

Пример:

Настроить счетчик импульсов для линии IO7 (должна быть настроена на вход) на срабатывание при переходе входного сигнала с низкого на высокий логический уровень:

запрос: $KE,IPL,EVT,I,7,SET,0
ответ: #IPL,EVT,SET,OK

Синтаксис 2: $KE,IPL,EVT,<Type>,<Line>,GET

Чтение текущего состяния настройки типа события для счетчика импульсов.

Ответ на запрос:

#IPL,EVT,<Type>,<Line>,<Value>

Параметры:

Type	–	Тип входной линии с которой ассоциирован данный счетчик(и): 'L' – оптоизолированные входные линии IN1 – IN6 'I' – дискретные линии общего назначения IO1 – IO8 настроенные на вход
Line	–	Номер счетчика для данного типа входной линии
Value	–	Счетчик будет реагировать на: 0 – переход уровня сигнала на входе с низкого на высокий 1 – с высокого на низкий 2 – оба варианта

Пример:

Считать текущее значение настройки для счетчика импульсов входной оптоизолированной дискретной линии IN2:

запрос: $KE,IPL,EVT,L,2,GET
ответ: #IPL,EVT,L,2,1

$KE,IPL,ENB

Управление настройкой включения / выключения функции счетчика импульсов.

Синтаксис 1: **$KE,IPL,ENB,<Type>,<Line>,SET,<Value>**

Установка настройки разрешения функции подсчета импульсов.

Параметры:

Type	–	Тип входной линии с которой ассоциирован данный счетчик(и): 'L' – оптоизолированные входные линии IN1 – IN6 'I' – дискретные линии общего назначения IO1 – IO8 настроенные на вход
Line	–	Номер счетчика для данного типа входной линии
Value	–	Функция счетчика импульсов: 0 –выключена (запрщена) 1 – включена (разрешена)

Ответ на запрос:

#IPL,ENB,SET,OK

Пример:

Включить функцию счетчка импульсов для дискретной оптоизолированной линии IN4:

запрос: $KE,IPL,ENB,L,4,SET,1
ответ: #IPL,ENB,SET,OK

Синтаксис 2: $KE,IPL,ENB,<Type>,<Line>,GET

Чтение текущего состяния настройки разрешения функции счетчика импульсов.

Ответ на запрос:

#IPL,ENB,<Type>,<Line>,<Value>

Параметры:

Type	–	Тип входной линии с которой ассоциирован данный счетчик(и): ‘L’ – оптоизолированные входные линии IN1 – IN6 ‘I’ – дискретные линии общего назначения IO1 – IO8 настроенные на вход
Line	–	Номер счетчика для данного типа входной линии
		Счетчик будет реагировать на:
Value	–	Функция счетчика импульсов: 0 –выключена (запрщена) 1 – включена (разрешена)

Пример:

Считать текущее значение настройки разрешения функции счетчика импульсов для линии общего назначения IO1 (линия должна быть настроена на вход):

запрос: $KE,IPL,ENB,I,1,GET
ответ: #IPL,EVT,I,1,0

В данный момент функция счетчика импульсов для линии IO1 выключена.

$KE,IPL,MEM

Управление настройкой периодического сохранения значений счетчиков импульсов в энергонезависимой памяти модуля.

Синтаксис 1: **$KE,IPL,MEM,<Type>,<Line>,SET,<Value>**

Установка настройки сохранения значений считчиков импульсов.

Параметры:

Type	–	Тип входной линии с которой ассоциирован данный счетчик(и): ‘L’ – оптоизолированные входные линии IN1 – IN6 ‘I’ – дискретные линии общего назначения IO1 – IO8 настроенные на вход
Line	–	Номер счетчика для данного типа входной линии
Value	–	Функция периодического сохранения: 0 – выключена (запрщена) 1 – включена (разрешена)

Ответ на запрос:

#IPL,MEM,SET,OK

Пример:

Включить функцию сохранения значений счетчика в память для дискретной оптоизолированной входной линии IN4:

запрос: $KE,IPL,MEM,L,4,SET,1
ответ: #IPL,MEM,SET,OK

Синтаксис 2: $KE,IPL,MEM,<Type>,<Line>,GET

Чтение текущего состояния настройки сохранения значений счетчика импульсов в энергонезависимую память модуля.

Ответ на запрос:

#IPL,MEM,<Type>,<Line>,<Value>

Параметры:

Type	–	Тип входной линии с которой ассоциирован данный счетчик(и): ‘L’ – оптоизолированные входные линии IN1 – IN6 ‘I’ – дискретные линии общего назначения IO1 – IO8 настроенные на вход
Line	–	Номер счетчика для данного типа входной линии
Value	–	Функция периодического сохранения: 0 – выключена (запрщена) 1 – включена (разрешена)

Пример:

Считать текущее значение настройки периодического сохранения значений счетчика импульсов в память модуля для линии общего назначения IO3 (линия должна быть настроена на вход):

запрос: $KE,IPL,MEM,I,3,GET
ответ: #IPL,MEM,I,3,1

В данный момент функция сохранения для линии IO3 включена.

$KE,IPL,VAL

Команды данной группы позволяют считывать и перезаписывать (устанавливать) значение счетчика импульсов.

Синтаксис 1: **$KE,IPL,VAL,<Type>,<Line>,GET**

Чтение текущего значения счетчка импульсов.

Ответ на запрос:

#IPL,VAL,<Type>,<Line>,<Value>

Параметры:

Type	–	Тип входной линии с которой ассоциирован данный счетчик(и): ‘L’ – оптоизолированные входные линии IN1 – IN6 ‘I’ – дискретные линии общего назначения IO1 – IO8 настроенные на вход
Line	–	Номер счетчика для данного типа входной линии
Value	–	Значение счечтика импульсов, целое число в диапазоне [0 - 4294967295]

Пример:

Запоросить текущее значение счетка импульсов дискретной оптоизолированной входной линии IN4:

запрос: $KE,IPL,VAL,L,4,GET
ответ: #IPL,VAL,L,4,5780

Синтаксис 2: **$KE,IPL,VAL,<Type>,<Line>,SET,<Value>**

Запись нового значения в счетчик импульсов.

Ответ на запрос:

#IPL,VAL,SET,OK

Параметры:

Type	–	Тип входной линии с которой ассоциирован данный счетчик(и): ‘L’ – оптоизолированные входные линии IN1 – IN6 ‘I’ – дискретные линии общего назначения IO1 – IO8 настроенные на вход
Line	–	Номер счетчика для данного типа входной линии
Value	–	Значение счечтика импульсов, целое число в диапазоне [0 - 4294967295]

Пример:

Установить значение счетчка импульсов для оптоизолированной входной линии IN4 в значение 500:

запрос: $KE,IPL,VAL,L,4,SET,500
ответ: #IPL,VAL,SET,OK

Синтаксис 3: **$KE,IPL,VAL,<Type>,<Line>,PP[,<Value>]**

Инкримент текущего значения счетчика импульсов.

Ответ на запрос:

#IPL,VAL,PP,OK

Параметры:

Type	–	Тип входной линии с которой ассоциирован данный счетчик(и): 'L' – оптоизолированные входные линии IN1 – IN6 'I' – дискретные линии общего назначения IO1 – IO8 настроенные на вход
Line	–	Номер счетчика для данного типа входной линии
Value	–	Опциональный параметр. Если не указан – значение счетчика увеличится на единицу. В противном случае значение *Value* будет прибавлено к текущему значению счетчка. Целое число в диапазоне [0 - 4294967295]

Пример 1:

Увеличить текущее значение счетчка импульсов для оптоизолированной входной линии IN4 на единицу:

запрос: $KE,IPL,VAL,L,4,PP
ответ: #IPL,VAL,PP,OK

К аналогичному результату приведет команда:

запрос: $KE,IPL,VAL,L,4,PP,1
ответ: #IPL,VAL,PP,OK

Пример 2:

Увеличить текущее значение счетчка импульсов для оптоизолированной входной линии IN4 на 200:

запрос: $KE,IPL,VAL,L,4,PP,200
ответ: #IPL,VAL,PP,OK

Синтаксис 4: **$KE,IPL,VAL,<Type>,<Line>,MM[,<Value>]**

Декримент текущего значения счетчика импульсов.

Ответ на запрос:

#IPL,VAL,MM,OK

Параметры:

Type	–	Тип входной линии с которой ассоциирован данный счетчик(и): ‘L’ – оптоизолированные входные линии IN1 – IN6 ‘I’ – дискретные линии общего назначения IO1 – IO8 настроенные на вход
Line	–	Номер счетчика для данного типа входной линии
Value	–	Опциональный параметр. Если не указан – значение счетчика уменьшится на единицу. В противном случае значение *Value* будет вычтено из текущего значения счетчка. Целое число в диапазоне [0 - 4294967295]

Пример 1:

Уменьшить текущее значение счетчка импульсов для оптоизолированной входной линии IN4 на единицу:

запрос: $KE,IPL,VAL,L,4,MM
ответ: #IPL,VAL,MM,OK

К аналогичному результату приведет команда:

запрос: $KE,IPL,VAL,L,4,MM,1
ответ: #IPL,VAL,MM,OK

Пример 2:

Уменьшить текущее значение счетчка импульсов для оптоизолированной входной линии IN4 на 200:

запрос: $KE,IPL,VAL,L,4,MM,200
ответ: #IPL,VAL,MM,OK

$KE,IPL,DIR

Управление настройкой направления счета (инкримент / дикримент).

Синтаксис 1: **$KE,IPL,DIR,<Type>,<Line>,SET,<Value>**

Установка настройки сохранения значений считчиков импульсов.

Параметры:

Type	–	Тип входной линии с которой ассоциирован данный счетчик(и): 'L' – оптоизолированные входные линии IN1 – IN6 'I' – дискретные линии общего назначения IO1 – IO8 настроенные на вход
Line	–	Номер счетчика для данного типа входной линии
Value	–	Направление счета: 0 – инкримент (+1) 1 – дикримент (-1)

Ответ на запрос:

#IPL,DIR,SET,OK

Пример:

Настроить счетчик для дискретной оптоизолированной входной линии IN4 таким образом что бы он дикириментировал свое значение при поступлении импульса:

запрос: $KE,IPL,DIR,L,4,SET,1
ответ: #IPL,DIR,SET,OK

Синтаксис 2: $KE,IPL,DIR,<Type>,<Line>,GET

Чтение текущего состяния настройки направления счета.

Ответ на запрос:

#IPL,DIR,<Type>,<Line>,<Value>

Параметры:

Type	–	Тип входной линии с которой ассоциирован данный счетчик(и): 'L' – оптоизолированные входные линии IN1 – IN6 'I' – дискретные линии общего назначения IO1 – IO8 настроенные на вход
Line	–	Номер счетчика для данного типа входной линии
Value	–	Направление счета: 0 – инкримент (+1) 1 – дикримент (-1)

Пример:

Считать текущее значение настройки направления счета для линии общего назначения IO3 (линия должна быть настроена на вход):

запрос: $KE,IPL,DIR,I,3,GET
ответ: #IPL,MEM,I,3,1

В данный момент счетчик импульсов для линии IO3 дикрементирует свое значение при поступлении нового импульса.

1-Wire

Команды управления и контроля аппаратными шинами 1-Wire.

$KE,OWI,PWR

Команда управляет питанием датчиков подключенных к шине 1-Wire, а именно включает или выключает подачу напряжения на клеммы *PA* (для канала ‘A’ 1-Wire) и *PB* (канал ‘B’ соответственно). Дополнительно, если питание выключается – линия данных (клеммы *1WA* или *1WB*) соответствующего канала 1-Wire переводится в состояние “на вход” для того чтобы устранить возможность “паразитного” питания датчиков от линии данных.

Синтаксис 1: **$KE,OWI,<Bus Id>,PWR,SET,<State>**

Ответ на запрос:

#OWI,PWR,SET,OK

Параметры:

Bus ID	–	Идентификатор шины 1-Wire. Для модуля Laurent-5 возможны два варианта: ‘A’ и ‘B’
State	–	ON – включить питание шины; OFF – выключить

Пример:

Выключить питание датчиков подключенных к каналу ‘B’ шины 1-Wire:

запрос: $KE,OWI,B,PWR,SET,OFF
ответ: #OWI,PWR,SET,OK

Синтаксис 2: **$KE,OWI,<Bus Id>,PWR,GET**

Ответ на запрос:

#OWI,<Bus Id>,PWR,<State>

Параметры:

Bus ID	–	Идентификатор шины 1-Wire. Для модуля Laurent-5 возможны два варианта: ‘A’ и ‘B’
State	–	1 – питание шины включено; 0 – выключено

Пример:

Запросить текущее состояние питания для канала ‘B’ шины 1-Wire:

запрос: $KE,OWI,B,PWR,GET
ответ: #OWI,B,PWR,0

$KE,OWI,MOD

Команда задает тип 1-Wire датчиков (температуры, влажности, iButton и т.д.) которые будут отслеживаться и обрабатываться модулем.

Синтаксис 1: **$KE,OWI,<Bus Id>,MOD,SET,<Type>**

Ответ на запрос:

#OWI,MOD,SET,OK

Параметры:

Bus ID – Идентификатор шины 1-Wire. Для модуля Laurent-5 возможны два варианта: ‘A’ и ‘B’

Type – Тип датчика для поиска и обработки. Возможны следующие значения:

Для канала ‘A’:
0 – нет датчиков (не использовать)
1 – датчики температуры Dallas DS18B20

Для канала ‘B’:
0 – нет датчиков (не использовать)
2 – датчик влажности и температуры DHT-11
3 – iButton (ключ Touch Memory) типа Dallas DS1990
4 – датчик влажности и температуры DHT-22

Пример:

Назначить iButton датчики DS1990 для поиска и обработки на 1-Wire канал ‘B’ модуля Laurent-5:

запрос: $KE,OWI,B,MOD,SET,3
ответ: #OWI,MOD,SET,OK

Синтаксис 2: $KE,OWI,<Bus Id>,MOD,GET

Ответ на запрос:

#OWI,<Bus Id>,PWR,<Type>

Параметры:

Bus ID	–	Идентификатор шины 1-Wire. Для модуля Laurent-5 возможны два варианта: ‘A’ и ‘B’
Type	–	Тип датчика для поиска и обработки. См. описание синтаксиса команды *SET* для подробностей

Пример:

Запросить текущий тип датчиков назначенных для канала ‘B’ шины 1-Wire:

запрос: $KE,OWI,B,MOD,GET
ответ: #OWI,B,MOD,3

1-Wire датчики температуры

Команды управления и контроля дтачиков температуры 1-Wire класса DS18B20.

$KE,TMP,SCAN

Команда запускает процесс сканирования шины 1-Wire (канал A) на предмет поиска подключенных к ней 1-Wire датчиков температуры DS18B20.

Синтаксис: $KE,TMP,SCAN

Ответ на запрос:

#TMP,SCAN,OK

Пример:

Запустить поиск датчиков 1-Wire подключенных к каналу ‘A’ (в модуле Laurent-5 есть два незаивисимых канала 1-Wire):

запрос: $KE,TMP,SCAN
ответ: #TMP,SCAN,OK

$KE,TMP,GET,NUM

Возвращает число обнаруженных 1-Wire датчиков температуры DS18B20 на канале ‘A’.

Синтаксис: $KE,TMP,GET,NUM

Ответ на запрос:

#TMP,NUM,<Value>

Параметры:

Value – Число обнаруженных датчиков 1-Wire подключенных к каналу ‘A’.

Пример:

Запросить число обнаруженных ранее датчиков на шине 1-Wire канала ‘A’:

запрос: $KE,TMP,GET,NUM
ответ: #TMP,NUM,5

Последний раз (после вызова $KE,TMP,SCAN) на шине было обнаружено 5 устройств.

$KE,TMP,ADR

Команда возвращает адрес ранее обнаруженного 1-Wire устройства на канале шины 1-Wire ‘A’.

Синтаксис: $KE,TMP,ADR,<SensorId>

Ответ на запрос:

#TMP,ADR,<SensorId>,<Value>

Параметры:

SensorId	–	Номер 1-Wire устройства. Может быть в диапазоне от 1 до NUM, где NUM – число обнарудженных устройств вызвращенное кпо команду $KE,TMP,GET,NUM
Value	–	8-ми байтный адрес 1-Wire устройства

Пример:

Запросить адрес обнаруженного 1-Wire устройства с ID = 2:

запрос: $KE,TMP,ADR,2
ответ: #TMP,ADR,2,40.20.81.87.8.0.0.186

$KE,TMP,VAL

Команда возвращает значение температуры в градусах Цельсия для указанного датчика температуры класса DS18B20 ранее обнаруженного на канале шины 1-Wire ‘A’.

Синтаксис: $KE,TMP,VAL,<SensorId>

Ответ на запрос:

#TMP,VAL,<SensorId>,<Value>

Параметры:

SensorId	–	Номер 1-Wire устройства. Может быть в диапазоне от 1 до NUM, где NUM – число обнарудженных устройств вызвращенное по команде $KE,TMP,GET,NUM
Value	–	Значение температуры в градусах Цельсия

Пример:

Запросить текущие показания датчика температуры класса DS18B20 с ID = 3:

запрос: $KE,TMP,VAL,3
ответ: #TMP,VAL,3,27.50

RFID & iButton

Команды управления и контроля работы декодера Wiegand данных от внешнего считывателя RFID а так же iButton.

$KE,WGN

Команда включает / выключает функционал декодера Wiegand данных от внешнего RFID считытвателя. Данные от считывателя поступают на линии ADC2 и ADC4 (использование этих каналов АЦП для задач измерения напряжения при включенном декодере Wiegand будет невозможным). Настройка сохраняется в энергонезависимой памяти.

Синтаксис 1: **$KE,WGN,SET,<Value>**

Параметры:

Value – 0 – выключить декодер данных Wiegand
1 – включить в режим Wiegand-26
2 – включить в режим Wiegand-42

Ответ на запрос:

#WGN,SET,OK

Пример:

Включить декодер Wiegand-42 данных от внешнего считывателя:

запрос: $KE,WGN,SET,2
ответ: #WGN,SET,OK

Синтаксис 2: **$KE,WGN,GET**

Возвращает текущее состояние декодера данных Wiegand (включено / выключено).

Ответ на запрос:

#WGN,<Value>

Параметры:

Value – 0 – выключить декодер данных Wiegand
1 – включить в режим Wiegand-26
2 – включить в режим Wiegand-42

Пример:

Запросить текущее состояние декодера:

запрос: $KE,WGN,GET
ответ: #WGN,0

В текущий момент времени декодер выключен.

$KE,WGN,TOUT

Команда задает значения временных таймаутов используемых алгоритмом чтения и обработки данных от внешнего RFID считывателя по интерфейсу Wiegand.

Синтаксис 1: **$KE,WGN,TOUT,SET,<T1>,<T2>**

Параметры:

T1	–	Время в милисекундах, допустимые знаяения [0 – 5000 мс]. Ранее накопленные данные в буфере приема Wiegand будут сброшены и начнентся набор нового сообщения с начала, если за интервал времени *T1* не поступало данных от считыватля. Значение по умолчанию – 500 мс.
T2	–	Время в милисекундах, допустимые знаяения [0 – 5000 мс]. Обработка и проверка (контрольная сумма) накопленных данных от считывателя будет начала не ранее чем через *T2* милисекунд с момента приема последнего информационного бита данных от считывателя. Значение по умолчанию – 300 мс.

Ответ на запрос:

#WGN,TOUT,SET,OK

Пример:

Изменить настройки таймаутов обработчика данных и сделать его более “реактивным”:

запрос: $KE,WGN,TOUT,SET,100,50
ответ: #WGN,TOUT,SET,OK

Синтаксис 2: $KE,WGN,TOUT,GET

Возвращает текущее значение внутренних таймаутов декодера данных Wiegand.

Ответ на запрос:

#WGN,TOUT,<T1>,<T2>

Параметры:

См. описание команды $KE,WGN,TOUT,SET

Пример:

Запросить текущее состояние таймаутов декодера:

запрос: $KE,WGN,TOUT,GET
ответ: #WGN,TOUT,500,300

В текущий момент времени декодер использует таймауты в 500 и 300 мс соответственно.

$KE,RFD

Команда управляет базой данных идентификаторов RFID и iButton ключей / меток хранимых в энергонезависимой памяти модуля.

Синтаксис 1: **$KE,RFD,GET,NUM**

Возвращает число сохраненных в базе RFID / iButton меток.

Ответ на запрос:

#RFD,NUM,<MAX_Num>,<Value>

Параметры:

MAX_Num – Емкость базы данных. Максимально возможное число меток для хранения

Value – Число валидных меток в базе данных

Пример:

Запросим число валидных меток:

запрос: $KE,RFD,GET,NUM
ответ: #RFD,NUM,250,17

Синтаксис 2: **$KE,RFD,ALL,DEL**

Удаляет все записи в базе данных.

Ответ на запрос:

#RFD,ALL,DEL,OK

Пример:

Полностью очистим базу данных RFID / iButton меток:

запрос: $KE,RFD,ALL,DEL
ответ: #RFD,ALL,DEL,OK

Синтаксис 3: **$KE,RFD,<Id>,SET,<Type>,<Value>**

Сохранение нового RFID / iButton идентификатора в базу данных.

Параметры:

Id	–	Номер ячейки в базе данных. Допустимые значения [1 - 1000] включительно. Значение -1 означет что данные будут записаны в первую свободную ячейку базы данных.
Type	–	Определяет в каком формате будут переданы данные: ‘H’ – в HEX форме, например 05A6F1 ‘D’ – в десятичной форме, числа разделены точкой, например: 5.166.241 ‘d’ – в десятичной форме, в виде одного числа (не более 3 байт) например 370417
Value	–	ID RFID / iButton ключа. 3-х байтный формат для RFID ключей (соответствует Wiegand-26) и 8-ми байтный формат для iButton

Ответ на запрос:

#RFD,SET,OK

Пример 1:

Имеем следующую RFID EM Marine карту доступа. Необходимо внести ее в базу данных модуля Laurent-5 (например, для последующей привязки к событиям CAT):

Идентификатор карты удобно сохранить в десятичном формате, именно так как он указан на самой карте:

запрос: $KE,RFD,-1,SET,d,4830263
ответ: #RFD,SET,OK

В данном примере карта была сохранена в первой свободной ячейке в базе данных. Можно сохранять данные и в HEX форме. Например, ID 4830263 в DEC виде соответствует запись 49B437 в HEX виде:

запрос: $KE,RFD,-1,SET,H,49B437
ответ: #RFD,SET,OK

Пример 2:

Был обнаружен ключ iButton (Touch Memory) с полным 1-Wire идентификатором (8 байт) 1.232.63.79.1.0.0.143 (dec) или он же но в HEX форме 01E83F4F0100008F (hex). Необходимо внести ее в базу данных в ячейку под номером 35.

запрос: $KE,RFD,35,SET,H,01E83F4F0100008F
ответ: #RFD,SET,OK

Аналогично можно подать команду вида:

запрос: $KE,RFD,35,SET,D,1.232.63.79.1.0.0.143
ответ: #RFD,SET,OK

Синтаксис 4: $KE,RFD,<Id>,GET,<Type>

Считывание идентификатора RFID карты из базы данных.

Ответ на запрос:

#RFD,<ID>,<Type>,<Len>,<Value>

Параметры:

Id	–	Номер ячейки в базе данных. Допустимые значения [1 - 50] включительно.
Type	–	Определяет в каком формате будут переданы данные: ‘D’ – в десятичной форме с разделитем в виде точками между байтами данных ‘H’ – в HEX форме ‘d’ – в десятичной форме в виде единого числа (не более 3 байт)
Value	–	ID RFID ключа, 3-х байтный формат (соответствует Wiegand-26) для RFID или 8 байт для iButton

Пример:

Считать идентификатор RFID карты сохраненной в ячейке базы данных под номером 23 в HEX виде:

запрос: $KE,RFD,23,GET,H
ответ: #RFD,23,H,3,45F6A4

Синтаксис 5: **$KE,RFD,<Id>,DEL**

Удаление записи из базы данных.

Ответ на запрос:

#RFD,DEL,OK

Параметры:

Id – Номер ячейки в базе данных. Допустимые значения [1 - 50] включительно.

Пример:

Удалить (очистить) запись в базе данных RFID под номером 8:

запрос: $KE,RFD,8,DEL
ответ: #RFD,DEL,OK

ModbusTCP

Команды управления и контроля работы ModbusTCP сервера.

$KE,MBS

Управление работой режима ModbusTCP сервера (включить / выключить).

Синтаксис 1: **$KE,MBS,SET,<State>**

Включение / выключение ModbusTCP сервера.

Параметры:

State – ON – включить ModbusTCP сервер
OFF – выключить

Ответ на запрос:

#MBS,SET,OK

Пример:

Включить ModbusTCP сервер:

запрос: $KE,MBS,SET,ON
ответ: #MBS,SET,OK

Синтаксис 2: **$KE,MBS,GET**

Запрос текущего состояния ModbusTCP сервера.

Ответ на запрос:

#MBS,<State>

Параметры:

State – 1 – ModbusTCP сервер включен
0 – выключен

Пример:

Запросить текущий статус ModbusTCP сервера:

запрос: $KE,MBS,GET
ответ: #MBS,1

В данный момент времени TCP клиент включен.

$KE,MBS,PRT

Управление настройкой ModbusTCP сервера – TCP порт к которому должны подключаться внешние ModbusTCP клиенты.

Синтаксис 1: **$KE,MBS,PRT,SET,<Value>**

Установка значения TCP порта сервера.

Параметры:

Value – Номер TCP порта, целое число [0 - 65535]. Значение по умолчанию – 502.

Ответ на запрос:

#MBS,PRT,SET,OK

Пример:

Установить TCP порт ModbusTCP сервера равным 650:

запрос: $KE,MBS,PRT,SET,650
ответ: #MBS,PRT,SET,OK

Синтаксис 2: $KE,MBS,PRT,GET

Запрос значения TCP порта ModbusTCP сервера.

Ответ на запрос:

#MBS,PRT,<Value>

Параметры:

Value – Номер TCP порта, целое число [0 - 65535]

Пример:

Запрос значения TCP порта ModbusTCP сервера:

запрос: $KE,MBS,PRT,GET
ответ: #MBS,PRT,502

Датчик влажности DHT-11

Команды управления и контроля показаний датчика влажности и температуры класса DHT-11.

$KE,HMD,VAL,GET

Получить текущий статус датчика влажности класса DHT-11 (подключен / неподключен / данные валидны) и измерения датчика (влажность, температура).

Синтаксис: $KE,HMD,VAL,GET

Ответ на запрос:

#HMD,VAL,<Connected>,<Valid>,<Humidity>,<Temperature>

Параметры:

Connected	–	Состояние датчика: 0 – датчик не подключен (не обнаружен) 1 – датчик подключен
Valid	–	Флаг валидности измерений: 0 – измерения не валидны (использовать нельзя) 1 – измерения валидны
Humidity	–	Относительная влажность в %
Temperature	–	Температура в градуса Цельсия

Пример:

Запустить текущее состояние и показания датчика влажности типа DHT-11 (должен быть подключен к шине 1-Wire, канал ‘B’ модуля):

запрос: $KE,HMD,VAL,GET
ответ: #HMD,VAL,1,1,35.0,26.0

В текущий момент времени датчик обнаружен, измерения корректные, влажность – 35%, температура +26 C°.

CAT Система CAT

Команды управления и контроля работой системы CAT.

$KE,CAT

Команда управления и настройки событий системы CAT.

Синтаксис 1: **$KE,CAT,ALL,SET,<State>**

Выполнение ряда операций (включение, выключение) для всех событий CAT за одно обращение.

Параметры:

State – OFF – выключить все CAT события
ON – включить

Ответ на запрос:

#CAT,ALL,SET,OK

Пример:

Выключить все созданные CAT события:

запрос: $KE,CAT,ALL,SET,OFF
ответ: #CAT,ALL,SET,OK

Синтаксис 2: **$KE,CAT,<Id>,SET,<State>**

Выполнение ряда операций (включение, выключение) для конкретного CAT события

Параметры:

Id	–	Номер CAT события, целое число в диапазоне [1 - 50] включительно
State	–	OFF – выключить событие ON – включить OPP – переключить в состоняие противоположенное текущему

Ответ на запрос:

#CAT,SET,OK

Пример:

Выключить событие CAT под номером 23:

запрос: $KE,CAT,23,SET,OFF
ответ: #CAT,SET,OK

Синтаксис 3: **$KE,CAT,<Id>,GET**

Текущий статус конкретного события CAT (включено / выключено).

Ответ на запрос:

#CAT,<Id>,<State>

Параметры:

Id – Номер CAT события, целое число в диапазоне [1 - 50] включительно

State – OFF – событие выключено
ON – включено

Пример:

Запросить текущий статус события CAT под номером 23:

запрос: $KE,CAT,23,GET
ответ: #CAT,23,OFF

В данный момент событие под номером 23 выключено.

Синтаксис 4: $KE,CAT,<Id>,DEL

Удалить конреткное CAT событие.

Параметры:

Id – Номер CAT события, целое число в диапазоне [1 - 50] включительно

Ответ на запрос:

#CAT,DEL,OK

Пример:

Удалить событие CAT под номером 23:

запрос: $KE,CAT,23,DEL
ответ: #CAT,DEL,OK

Синтаксис 5: **$KE,CAT,ALL,DEL**

Команда позволяет за одно обращение удалить из энергонезависимой памяти все CAT события.

Ответ на запрос:

#CAT,ALL,DEL,OK

Пример:

Удалить все события CAT:

запрос: $KE,CAT,ALL,DEL
ответ: #CAT,ALL,DEL,OK

$KE,CAC

Команда управления считчиками событий системы CAT.

Синтаксис 1: **$KE,CAC,ALL,RST**

Сбросить счетчики срабатываний всех событий CAT.

Ответ на запрос:

#CAC,ALL,RST,OK

Пример:

Обнулить счетчики срабатываний всех событий CAT:

запрос: $KE,CAC,ALL,RST
ответ: #CAT,ALL,RST,OK

Синтаксис 2: **$KE,CAC,<Id>,GET**

Текущее значение считчика срабатываний конкретного события CAT.

Ответ на запрос:

#CAC,<Id>,<Value>

Параметры:

Id – Номер CAT события, целое число в диапазоне [1 - 50] включительно

Value – Счетчик срабатываний, целое число

Пример:

Запросить значение счетчика срабатываний события CAT под номером 17:

запрос: $KE,CAC,17,GET
ответ: #CAC,17,100

На текущий момент времени CAT событие под номером 17 сработало 100 раз.

Синтаксис 3: **$KE,CAC,<Id>,RST**

Обнулить счетчик срабатываний для конкретного CAT события.

Параметры:

Id – Номер CAT события, целое число в диапазоне [1 - 50] включительно

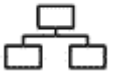
Ответ на запрос:

#CAT,RST,OK

Пример:

Обнулить счетчик срабатываний для события под номером 17:

запрос: $KE,CAC,17,RST
ответ: #CAT,RST,OK



TCP клиент

Команды управления и контроля работы режима TCP клиента.

$KE,BTCPC

Управление работой режима TCP клиента (включить / выключить).

Синтаксис 1: **$KE,BTCPC,SET,<State>**

Включение / выключение TCP клиента.

Параметры:

State – ON – включить TCP клиент
OFF – выключить

Ответ на запрос:

#BTCPC,SET,OK

Пример:

Включить TCP клиент:

запрос: $KE,BTCPC,SET,ON
ответ: #BTCPC,SET,OK

Синтаксис 2: **$KE,BTCPC,GET**

Запрос текущего состояния TCP клиента.

Ответ на запрос:

#BTCPC,<State>

Параметры:

State – 1 – TCP клиент включен
0 – выключен

Пример:

Запросить текущий статус TCP клиента:

запрос: $KE,BTCPC,GET
ответ: #BTCPC,1

В данный момент времени TCP клиент включен.

$KE,BTCPC,IP

Управление настройкой TCP клиента – IP адрес удаленной машины (TCP сервер) к которой должен подключаться TCP клиент модуля.

Синтаксис 1: **$KE,BTCPC,IP,SET,<Value>**

Установка значения IP адреса удаленной машины.

Параметры:

Value – IP адрес в формате X.X.X.X (в качестве X могут быть использованы числа от 0 до 255).

Ответ на запрос:

#BTCPC,IP,SET,OK

Пример:

Установить IP адрес удаленной машины к которой должен будет подключаться TCP клиент равным 192.168.1.200:

запрос: $KE,BTCPC,IP,SET,192.168.1.200
ответ: #BTCPC,IP,SET,OK

Синтаксис 2: **$KE,BTCPC,IP,GET**

Запрос значения IP адреса удаленной машины.

Ответ на запрос:

#BTCPC,IP,<Value>

Параметры:

Value – IP адрес в формате X.X.X.X (в качестве X могут быть использованы числа от 0 до 255).

Пример:

Запрос значения IP адреса удаленной машины:

запрос: $KE,BTCPC,IP,GET
ответ: #BTCPC,IP,192.168.1.200

$KE,BTCPC,PRT

Управление настройкой TCP клиента – TCP порт удаленной машины (TCP сервер) к которой должен подключаться TCP клиент модуля.

Синтаксис 1: **$KE,BTCPC,PRT,SET,<Value>**

Установка значения TCP порта удаленной машины.

Параметры:

Value – Номер TCP порта, целое число [0 - 65535]

Ответ на запрос:

#BTCPC,PRT,SET,OK

Пример:

Установить TCP порт удаленной машины к которому должен подключаться TCP клиент равным 8003:

запрос: $KE,BTCPC,PRT,SET,8003
ответ: #BTCPC,PRT,SET,OK

Синтаксис 2: $KE,BTCPC,PRT,GET

Запрос значения TCP порта удаленной машины.

Ответ на запрос:

#BTCPC,PRT,<Value>

Параметры:

Value – Номер TCP порта, целое число [0 - 65535]

Пример:

Запрос значения TCP порта удаленной машины к которому должен подключаться TCP клиент:

запрос: $KE,BTCPC,PRT,GET
ответ: #BTCPC,PRT,8003

GSM

Настройка и управление работой GSM модема поддержаного в модуле Laurent-5G. GSM функционал не поддерживается модулем Laurent-5.

$KE,PIN

Команда задает PIN код SIM карты который следует использовать при регистрации в сети GSM. Модуль в автоматическом режиме производит запуск и инициализацию GSM модема. Для этого ему необходимо знание PIN кода. Данные сохраняются в энергонезависимой памяти модуля.

Синтаксис 1: **$KE,PIN,SET,<Value>**

Установка значения SIM PIN кода.

Параметры:

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

Ответ на запрос:

#PIN,SET,OK

Пример:

Установить SIM PIN код 1234:

запрос: $KE,PIN,SET,1234
ответ: #PIN,SET,OK

Синтаксис 2: **$KE,PIN,GET**

Запрос текущего значения PIN кода SIM карты.

Ответ на запрос:

#PIN,<Value>

Параметры:

Value – PIN код SIM карты. Длина – 4 символа. Если PIN код еще не был задан, модуль выдает ответ в виде: #PIN,NOTSET

Пример:

запрос: $KE,PIN,GET
ответ: #PIN,1234

$KE,GMD

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

Синтаксис 1: **$KE,GMD,SET,<Value>**

Установка режима работы GSM функционала.

Параметры:

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

0: GSM функционал выключен. GSM модем обесточен, прием / отправка звонков и SMS невозможна.

1: GSM функционал работаем в автоматическом режиме. Модуль самостоятельно реестрируется в сети GSM и ожидает поступления входящих SMS / звонков.

Ответ на запрос:

#GMD,SET,OK

Пример:

Настроить GSM модем в автоматический режим работы:

запрос: $KE,GMD,SET,1
ответ: #GMD,SET,OK

Синтаксис 2: $KE,GMD,GET

Запрос текущего значения режима работы GSM функционала.

Ответ на запрос:

#GMD,<Value>

Параметры:

Value – См. описание команды $KE,GMD,SET

$KE,GSM,RST

Перезапуск GSM функционала. При подаче команды происходит физическое выключение GSM модема (отключение питания), повторное включение и регистрация в сети GSM.

Синтаксис: $KE,GSM,RST

Ответ на запрос:

#GSM,RST,OK

Пример:

Сбросить GSM функционал (перезагрузить GSM модем):

запрос: $KE,GSM,RST
ответ: #GSM,RST,OK

$KE,GSM,STAT

Возвращает текущий статус работы GSM функционала / модема.

Синтаксис: $KE,GSM,STAT

Ответ на запрос:

#GSM,STAT,<State>.<ErrCode>

Параметры:

State	–	Текущее состояние работы GSM модема. Целое число, возможные значения: 0 – GSM модем выключен 1– идет процесс настройки GSM модема / регистрации в сети 2 – модуль успешно разеристировался в GSM сети; готов к приему и отправке звонков / SMS
ErrCode	–	Код обнаруженной ошибки. Целое число, возможные значения: 0 – ошибок нет 1 – зарезервировано 2 – не установлена SIM карта 3 – указанный PIN код не соответствует SIM карте 4 – GSM модем требует введения PUK кода 5 – иная причина

Пример:

Запросить текущий статус GSM:

запрос: $KE,GSM,STAT
ответ: #GSM,STAT,2.0

GSM модем успешно настроен, выполнена регистрация в сети. Ошибок не обнаружено.

$KE,GSM,ICAL

Настройка позволяет блокировать прием и обработку звонков от телефонных номеров отличных от указанных в “белом” списке (см. команду *$KE,PHN*).

Синтаксис 1: **$KE,GSM,ICAL,SET,<Value>**

Установка режима (фильтра) входящих звонков.

Параметры:

Числовой параметр, определяющий режим работы фильтра по звонкам. Допустимые значения:

Value – 0: принимать звонки (снимать трубку) для любых входящих звонков
1: всегда сбрасывать входящий звонок для всех номеров
2: поднимать трубку для “белых” номеров, для отсальных - сбрасывать

Ответ на запрос:

#GMD,ICAL,SET,OK

Пример:

Настроить GSM модем так что бы принимать звонки (поднимать трубку) только для телефонных номеров указанных в “белом” списке:

запрос: $KE,GSM,ICAL,SET,2
ответ: #GSM,ICAL,SET,OK

Синтаксис 2: $KE,GSM,ICAL,GET

Запрос текущего значения настройки фильтра по входящим звонкам.

Ответ на запрос:

#GSM,ICAL,<Value>

Параметры:

Value – См. описание команды $KE,GSM,ICAL,SET

$KE,GSM,ISMS

Настройка позволяет блокировать обработку входящих SMS от телефонных номеров отличных от указанных в “белом” списке (см. команду *$KE,PHN*).

Синтаксис 1: **$KE,GSM,ISMS,SET,<Value>**

Установка режима (фильтра) входящих SMS.

Параметры:

Value – Числовой параметр, определяющий режим работы фильтра по SMS. Допустимые значения:

0: всегда обрабатывать SMS от любых телефонных номеров
1: всегда игнорировать входящие SMS
2: обрабатывать SMS только от “белых” номеров

Ответ на запрос:

#GMD,ISMS,SET,OK

Пример:

Настроить GSM модем так что бы обрабатывать SMS только для телефонных номеров указанных в “белом” списке:

запрос: $KE,GSM,ISMS,SET,2
ответ: #GSM,ISMS,SET,OK

Синтаксис 2: $KE,GSM,ISMS,GET

Запрос текущего значения настройки фильтра по входящим SMS.

Ответ на запрос:

#GSM,ISMS,<Value>

Параметры:

Value – См. описание команды $KE,GSM,ISMS,SET

$KE,PHN

Управление списоком “белых” телефонных номеров хранимых в энергонезависимой памяти модуля.

Синтаксис 1: **$KE,PHN,<Id>,<Mode>,SET,<Value>**

Добавление нового телефонного номера в базу.

Параметры:

Id	–	Номер ячейки в базе данных. Допустимые значения [1 - 32] включительно. Значение 0 означет что данные будут записаны в первую свободную ячейку базы данных.
Type	–	Определяет в каком формате будут переданы данные: ‘H’ – в HEX форме ‘C’ – в строковом формате.
Value	–	Телефонный номер. Допустимые символы: 0-9, ‘+’. В том случае если команда передается через HTTP GET запрос – следует избежать использование символа ‘+’ в строков формате либо перейти на использование HEX формы

Ответ на запрос:

#PHN,SET,OK

Пример:

Добавить телефонный номер +7-916-123-45-67 в базу “белых” номеров в первую свободную ячейку:

запрос: $KE,PHN,0,C,SET,+79161234567
ответ: #PHN,SET,OK

Тот же самый номер, но указанный в HEX формате (может быть полезным в том случае если тело команды нужно передавать в рамках URL HTTP запроса в котором символ ‘+’ недопустим):

запрос: $KE,PHN,0,H,SET,2B37393137313233343536 37
ответ: #PHN,SET,OK

Синтаксис 2: **$KE,PHN,<Id>,<Mode>,GET**

Запрос текущего телефонного номера из базы в ячейке *Id* в строковом или HEX формате (*Mode*).

Ответ на запрос:

#PHN,<Id>,<Value>

Параметры:

Id	–	Номер ячейки в базе данных. Допустимые значения [1 - 32] включительно.
Type	–	Определяет в каком формате будут выданы данные: ‘H’ – в HEX форме ‘C’ – в строковом формате.
Value	–	Телефонный номер. Допустимые символы: 0-9, ‘+’. В том случае если ячейка пуста (телефонный номер не указан), будет возвращено значение NOTSET

Синтаксис 3: $KE,PHN,DEL,ALL

Удалить все записи из базы “белых” телефонных номеров.

Ответ на запрос:

#PHN,DEL,ALL,OK

Синтаксис 4: $KE,PHN,<Id>,DEL

Удаление телефонного номра из “белой” базы в ячейке *Id*

Ответ на запрос:

#PHN,DEL,OK

Параметры:

Id – Номер ячейки в базе данных. Допустимые значения [1 - 32] включительно.

$KE,RNG

Выполнение звонка на указанный телефонный номер из “белого” списка. Команада может быть использована как реакция модуля в системе САТ на некоторое событие.

Синтаксис: $KE,RNG,DO,<Id>

Ответ на запрос:

#RNG,<State>

Параметры:

Id	–	Номер ячейки в базе “белых” номеров. Допустимые значения [1 - 32] включительно.
State	–	*DO,OK* – задание на совершение звонка поставлено в очередь *ERR,NO_FREE_PLACE* – закончилась память в очереди задач на совершение звонка *ERR,BAD_PHN* – звонок на указанный номер не возможен, т.к. он либо пуст либо невалиден

Пример:

Совершить звонок на “белый” номер записанный в базе под ID = 4:

запрос: $KE,RNG,DO,4
ответ: #RNG,DO,OK

$KE,SMS

Отправка SMS на указанный телефонный номер из “белого” списка. Команада может быть использована как реакция модуля в системе CAT на некоторое событие.

Синтаксис: $KE,SMS,SND,<Id>,<Mode>,<Msg>

Ответ на запрос:

#SMS,<State>

Параметры:

Id	–	Номер ячейки в базе данных “белых” номеров. Допустимые значения [1 - 32] включительно.
Mode	–	Определяет, в каком формате представлен текст SMS: ‘H’ – в HEX форме ‘C’ – в строковом формате
Msg	–	Текст сообщения для отправки. Максимальная длина – 96 символов. Команда поддерживает работу с Динамическими переменными
State	–	*SND,OK* – задание на отправку SMS поставлено в очередь *ERR,NO_FREE_PLACE* – закончилась память в очереди задач на отправку SMS *ERR,BAD_PHN* – отправка SMS на указанный номер не возможна, т.к. он либо пуст либо невалиден

Пример 1:

Отправить SMS с текстом “ Hello!” на “белый” номер, записанный в базе под ID = 2:

запрос: $KE,SMS,SND,2,C,Hello!
ответ: #SMS,SND,OK

Пример 2:

Отправить SMS с динамическими показаниями (сформированными на момент отправки сообщения) датчика влажности класса DHT-11 и текущим временем / датой на “белый” номер, записанный в базе под ID = 1:

запрос: $KE,SMS,SND,1,C,Vlaznost=~Dh1~ ~LF~~DT~ ~TM~
ответ: #SMS,SND,OK

При этом на указанный номер сотового телефона придет сообщение следующего вида (показания влажности, дата и время из RTC будут подставлены в текст “на лету” в момент отправки):

Выдача / передача данных

Команды управления и настройки выдачи данных и информационных Ke-сообщений.

$KE,PUT

Отправка произвольных данных через указанный интерфейс (TCP сервер, TCP клиент, RS-232 порт).

Синтаксис: $KE,PUT,<Interface>,<Mode>,<Data>

Ответ на запрос:

#PUT,OK,<Len> - команда сформирвоана верно, данные поставлены в очередь на отправку

Параметры:

Interface	–	Буквенный идентификатор интерфейса, в который нужно отправить данные: ‘S’ – TCP сервер (доступынй по умолчанию на TCP порту 2424) ‘C’ – TCP клиент ‘U’ – порт RS-232
Mode	–	Формат передаваемыхданных: ‘C’ – ASCII ‘H’ – HEX
Data	–	Данные для отправки. Максимальная длина - не более 200 символов. Команда поддерживает работу с Динамическими переменными в случае использования текстовой строки в ASCII формате
Len	–	Число фактически отправленных байт данных в указанный интерфейс

Пример 1:

Отправить в текущее соединение к TCP серверу модуля строку ‘ALARM!’ в HEX виде, в конце строки добавить символы возврата каретки и перехода на новую строку (<CR><LF>):

запрос: $KE,PUT,S,H,414C41524D210D0A
ответ: #PUT,OK,8

Пример 2:

Отправить в порт TCP клиента строку ‘Hello’

запрос: $KE,PUT,C,C,Hello!
ответ: #PUT,OK,6

Пример 3:

Отправить в порт TCP сервера строку с динамическим временем и сотоянием 1-го реле:

запрос: $KE,PUT,S,C,Date=~DT~, Time=~TM~, RELE_1: ~RL1~
ответ: Date=2020.09.16, Time=11:17:52, RELE_1: 0
#PUT,OK,43

$KE,MSG

Управление выдачей информационных сообщений.

Синтаксис 1: **$KE,MSG,<Interface>,<Name>,SET,<State>**

Включение / выключение сообщения для конкретного интерфейса.

Параметры:

Interface	–	Буквенный идентификатор интерфейса: ‘S’ – TCP сервер (доступынй по умолчанию на TCP порту 2424) ‘C’ – TCP клиент ‘U’ – порт RS-232
Name	–	Текстовое имя сообщения. См. раздел “KE сообщения”
State	–	OFF – выключить выдачу сообщения в указанный порт (интерфейс) ON – разрешить (включить)

Ответ на запрос:

#MSG,SET,OK

Пример:

Включим выдачу сообщений о событиях CAT в порт TCP клиента:

запрос: $KE,MSG,C,ECAT,SET,ON
ответ: #MSG,SET,OK

Синтаксис 2: **$KE,MSG,<Interface>,<Name>,GET**

Запрос текущего состояния настройки выдачи сообщения в интерфес (порт).

Ответ на запрос:

#MSG,<Interface>,<Name>,<State>

Параметры:

Interface	–	Буквенный идентификатор интерфейса: ‘S’ – TCP сервер (доступынй по умолчанию на TCP порту 2424) ‘C’ – TCP клиент ‘U’ – порт RS-232
Name	–	Текстовое имя сообщения. См. раздел “KE сообщения”
State	–	OFF – выдача сообщения запрещена ON – разрешена

Пример:

Запросим текущее состояние настройки по выдаче сообщения ‘EIOI’ в порт RS-232:

запрос: $KE,MSG,U,EIOI,GET
ответ: #MSG,U,EIOI,OFF

Выдача сообщения ‘EIOI’ в порт RS-232 выключена.

$KE,URL

“Выполнение” URL ссылки на внешний адрес модулем. При вызове данной команды модуль используя выделенный для этого HTTP клиент произведет подключение по указанному IP, передаст необходимые HTTP заголовки на удаленный сервер для выполнения указанной URL ссылки (команды).

Синтаксис: $KE,URL,RUN,<IP>,<Port>,<Cmd>

Параметры:

IP	–	IP адрес удаленного сервера
Port	–	TCP порт удаленного сервера. Обычно равен 80 (порт HTTP по умолчанию)
State	–	URL строка которую необходимо передать на сервер. Максимальная длина 100 символов. Команда поддерживает работу с Динамическими переменными

Ответ на запрос:

#URL,RUN,OK – информирует о том, что синтаксис команды верный, задача поставлена на очередь в обработку.

#URL,RUN,NOSLOTS – закончилось место в очереди на обработку обращений. Длина очереди 15 элементов.

Пример 1:

На текущем модуле (IP = 192.168.0.101) необходимо с помощью Ke-команды вызвать обращение по HTTP (вызвать URL ссылку) на другой модуль Laurent-5 (например, IP = 192.168.0.200) и включить на нем 3-е реле:

http://192.168.0.200/cmd.cgi?psw=Laurent&cmd=REL,3,1

Используя команду $KE,URL на первом модуле следующего вида можно выполнить необходимую операцию:

запрос: $KE,URL,RUN,192.168.0.200,80,cmd.cgi?psw=Laurent&cmd=REL,3,1
ответ: #URL,RUN,OK

Пример 2:

На текущем модуле (IP = 192.168.0.101) необходимо с помощью Ke-команды вызвать обращение по HTTP (вызвать URL ссылку) на внешний WEB сервер (например, IP = 192.168.0.200) и передать ему текущее состояние 2-го реле с помощью динамической переменной.

Предположим, URL ссылка для случая включенного реле выглядит следующим образом:

http://192.168.0.200/process.php?rele=1

Используя команду $KE,URL на первом модуле следующего вида можно выполнить необходимую операцию. Вместо текста *~RL2~* модуль “на лету” подставит текущее состояние (включено / выключено) RELE_2:

запрос: $KE,URL,RUN,192.168.0.200,80, process.php?rele=~RL2~
ответ: #URL,RUN,OK

Ke-скрипты

Команды управления Ke-скриптами – наборы Ke-команд и служебных команд временной задержки хранимые в энергонезависимой памяти модуля.

$KE,SCR,RUN

Синтаксис: **$KE,SCR,<SlotID>,RUN[,Cnt]**

Включение выполнения Ke-скрипта хранимого в базе под ID = *SlotID* указанное число раз (*Cnt*).

Параметры:

SlotID	–	Номер ячейки в базе данных в которой хранится данных скрипт. Допустимые значения [1 - 10] включительно.
Cnt	–	Опциональный параметр. Если не указан – скрипт будет выполнен единожды. В противном случае, скрипт будет выполнен *Cnt* число раз. Значение 0 означает что скрипт будет выполняться бесконечно (зациклен) до тех пор пока он не будет остановлен командой *$KE,SCR,STOP* или ресетом модуля.

Ответ на запрос:

#SCR,RUN,OK	-	Скрипт успешно запущен на выполнение
#SCR,RUN,ERR	-	Выполнние не возможно по причине некорректности данных в скрипте

Пример 1:

Включим выполнение Ke-скрипта с ID=1 с повтором 10 раз:

запрос: $KE,SCR,1,RUN,10
ответ: #SCR,RUN,OK

Пример 2:

Включим выполнение Ke-скрипта с ID=3 с бесконечным повтором (зацикливание):

запрос: $KE,SCR,1,RUN,0
ответ: #SCR,RUN,OK

$KE,SCR,STOP

Синтаксис: **$KE,SCR,<SlotID>,STOP**

Команда останавливает выполнение Ke-скрипта хранимого в базе под ID = *SlotID*.

Параметры:

SlotID – Номер ячейки в базе данных в которой хранится данных скрипт. Допустимые значения [1 - 10] включительно.

Ответ на запрос:

#SCR,STOP,OK

Пример:

Остановим выполнение Ke-скрипта с ID=6:

запрос: $KE,SCR,6,STOP
ответ: #SCR,STOP,OK

KE,SCR,DEL

Синтаксис: **$KE,SCR,<SlotID>,DEL**

Команда останавливает выполнение Ke-скрипта и удаляет его из энергонезависимой памяти.

Параметры:

SlotID – Номер ячейки в базе данных в которой хранится данных скрипт. Допустимые значения [1 - 10] включительно.

Ответ на запрос:

#SCR,DEL,OK

Пример:

Удалим Ke-скрипт с ID=6:

запрос: $KE,SCR,6,DEL
ответ: #SCR,DEL,OK

Безопастность, права доступа

Команды управления и контроля безопастности и режима доступа к интерфейсам модуля.

$KE,PSW,SET

С помощью команды можно ввести пароль доступа для разблокировки командного интерфейса модуля.

Синтаксис: $KE,PSW,SET,<Password>

Параметры:

Password – Пароль для доступа к модулю, не более 9 символов

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 и в качестве пароля доступа к Web-интерфейсу. Новый пароль сохраняется в энергонезависимой памяти.

Синтаксис: $KE,PSW,NEW,<NewPassword>

Параметры:

NewPassword – Новый пароль, длинной не более 9 символов. Можно использовать символы 0-9, a-z, A-Z.

Ответ на запрос:

#PSW,NEW,OK – новый пароль успешно установлен

Пример:

Установить новый пароль “*SimSim*”:

запрос: $KE,PSW,NEW,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,PSW,GET

Запрос текущего значения пароля.

Синтаксис: $KE,PSW,GET

Ответ на запрос:

#PSW,<*PasswordLength*>,< *Password* >

Параметры:

Password – Пароль для доступа к модулю

PasswordLength – Длина пароля

Пример:

Запросим текущий пароль модуля:

запрос: $KE,PSW,GET
ответ: #PSW,7,Laurent

$KE,PSW,BLK

Команда блокирует командный интерфейс (необходимо будет снова указать пароль), если ранее он был разблокирован командой $KE,PSW,SET.

Синтаксис: $KE,PSW,BLK

Ответ на запрос:

#PSW,BLK,OK – командный интерфейс заблокирован; для его разблокировки необходимо подать команду $KE,PSW,SET

$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 интерфейсу и Web-серверу 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 текущего 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 порты для управления модулем (TCP сервер, по умолчанию 2424), web-интерфейса (по умолчанию 80) и порт интерфейса TCP-2-COM (по умолчанию 2525). 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 – порт TCP сервера, 1 – TCP-2-COM, 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 для указанного типа интерфейса.

Синтаксис: $KE,PRT,<Port Type>,GET

Параметры:

Port Type – 0 – порт TCP сервера, 1 – TCP-2-COM, 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

Сетевые настройки

Команды управления и контроля сетевыми настройками.

$KE,IP,SET

Команда позволяет установить статический 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:

IpAdress – 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 или другой командный интерфейс.

$KE,IP,GET

Возвращает текущий IP адрес модуля.

Синтаксис: $KE,IP,GET

Ответ на запрос:

#IP,<IpAdress>

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 или другой командный интерфейс.

$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 или другой командный интерфейс.

$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,NBN,SET

Команда позволяет установить символическое текстовое имя NetBIOS Name Service (NBNS). Параметр сохраняется в энергонезависимой памяти. Изменения вступают в силу после перезагрузки модуля (команда *$KE,RST* или сброс питания).

Синтаксис: $KE,NBN,SET,<Name>

Параметры:

Name – Текстовое имя длиной от 1 до 15 символов. Допустимые символы: a-z, A-Z, 0-9 и знак ‘-’. Недопустимо использование двух знаков ‘-’ подряд а так же использование его в началае и конце имени.

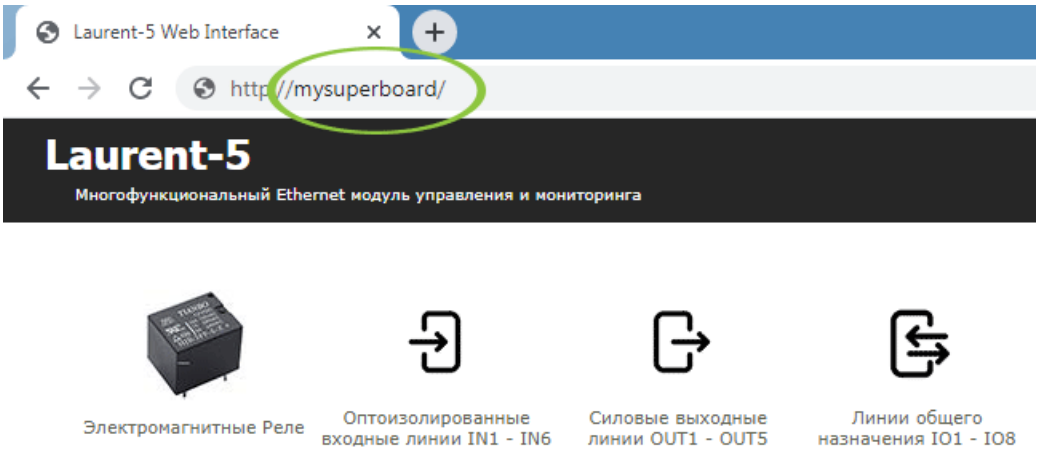
Ответ на запрос:

#NBN,SET,OK

Пример:

Установить значение NetBIOS Name в “mysuperboard”

запрос: $KE,NBN,SET,mysuperboard
ответ: #NBN,SET,OK



$KE,NBN,GET

Возвращает текущее значение NetBIOS Name.

Синтаксис: $KE,NBN,GET

Ответ на запрос:

#NBN,<Name>

Пример:

Получить текущее значение NetBIOS Name модуля:

запрос: $KE,NBN,GET
ответ: #NBN,mysuperboard

Дополнительные **настройки**

Дополнительные сервисные и служебные команды управления различными настройками и режимами работы модуля.

$KE,DZG

Настройка системы программного подавления дребезга контактов для входных оптоизолированных линий IN1 – IN6 и линий общего назначения IO1 – IO8 настроенных на вход. Настройка сохраняется с энергонезависимой памяти.

Синтаксис 1: **$KE,DZG,SET,<Type>,<Line>,<Value>**

Параметры:

Type	–	Тип входной линии: ‘L’ – оптоизолированные входные линии IN1 – IN6 ‘I’ – дискретные линии общего назначения IO1 – IO8 настроенные на вход
Line	–	Номер линии, [1-6] для типа ‘L’ (оптоизолированная входная линия) и [1-8] для типа ‘I’ (линии общего назначения)
Value	–	Число 25 мс интервалов задающих длительность подавления помех. Именно это время в миллисекундах будет необходимо системе подавления для принятия решения о наличии фиксированного уровня сигнала и отбраковки случайных “дребезгов”. Допустимые значения – [0 - 255] интервалов по 25 мс что соответствует времени подавления от 0 до 6,375 с. Значение 0 полностью отключает систему подавления дребезга контактов.

Ответ на запрос:

#DZG,SET,OK

Пример:

Установить постоянную времени подавителя дребеза контактов в 200 мс для входной оптоизолированной линии IN_5 (8 х 25 мс = 200 мс):

запрос: $KE,DZG,SET,L,5,8
ответ: #DZG,SET,OK

Синтаксис 2: **$KE,DZG,GET,<Type>,<Line>**

Команда возвращает текущее значение постоянной времени системы подавления дребезга контактов для указанной линии IN / IO.

Ответ на запрос:

#DZG,<Type>,<Line>,<Value>

Параметры:

См. описание для команды $KE,DZG,SET

Пример:

Считать текущее значение постоянной времени подавления дребезга контактов для линии IN_5:

запрос: $KE,DZG,GET,L,5
ответ: #DZG,L,5,8

Система подавления включена. Постоянная времени – 8 x 25 мс = 200 мс.

$KE,VAR

Команда управления пользовательскими переменными значения которых могут использоваться как условия для срабатывания системы САТ. Значения пользовательских переменных могут быть сохранены в энергонезависимой памяти.

Синтаксис 1: **$KE,VAR,<Id>,SET,<Value>[,S]**

Параметры:

Id	–	Номер пользовательской переменной VAR1 – VAR10. Может быть в диапазоне [1 - 10] включительно.
Value	–	Значение для присвоения переменной. Число с плавающей точкой.
N	–	Необязательный параметр, символ ‘S’. Если указан – то присвоенное значение для переменной VAR будет сохранено в энергонезависимой памяти модуля.

Ответ на запрос:

#VAR,SET,OK

Пример 1:

Установить переменной VAR3 значение 167, изменение сохранить в энергонезависимой памяти:

запрос: $KE,VAR,3,SET,167,S
ответ: #VAR,SET,OK

Пример 2:

Установить переменной VAR3 значение -2003.45, изменение не сохранять в энергонезависимой памяти:

запрос: $KE,VAR,3,SET,-2003.45
ответ: #VAR,SET,OK

Синтаксис 2: **$KE,VAR,<Id>,GET**

Команда возвращает текущее значение пользовательской перемнной.

Ответ на запрос:

#VAR,<Id>,<Value>

Параметры:

Id	–	Номер пользовательской переменной VAR1 – VAR10. Может быть в диапазоне [1 - 10] включительно.
Value	–	Значение переменной. Число с плавающей точкой

Пример:

Считать текущее значение пользовательской переменной VAR2:

запрос: $KE,VAR,2,GET
ответ: #VAR,2,0

Текущее значение для VAR2 – 0.

$KE,SAV

Управление режимом автоматического сохранения состояний аппаратных ресурсов в энергонезависимой памяти и их восстановления при сбросе / ресете модуля. Если режим включен – при каждом изменении состояния выбранных аппаратных ресурсов (реле, силовые выходные линии, ШИМ каналы, GPIO настроенные на “выход”) их текущее значение будет сохранено в энергонезависимой памяти модуля и автоматически восстановлено в случае сброса питания модуля.

Синтаксис 1: **$KE,SAV,<HW_Resource>,SET,<Value>**

Включение / выключение режима сохранения состояний аппартных ресурсов. Имеется возможность независимо управлять работой режима для каждого из поддерживаемых ресурсов.

Параметры:

HW_Resource – Тип аппаратного ресурса. Возможные значения:
REL – электромагнтыне реле Rele1 – Rele4
OUT – силовые выходные линии OUT_1 - OUT_5
GIO – слаботочные “выходные” линии IO1 – IO8
PWM – каналы ШИМ PWM_1 – PWM_4

Value – Состояние режима. *ON* – включить, *OFF* – выключить.

Ответ на запрос:

#SAV,SET,OK

Пример 1:

Включить режим автоматического сохранения и восстановления состояния выходных силовых линий OUT:

запрос: $KE,SAV,OUT,SET,ON
ответ: #SAV,SET,OK

Синтаксис 2: $KE,SAV,<HW_Resource>,GET

Возвращает текущее состояние системы автоматического сохранения значений для аппаратного ресурса *HW_Resource* .

Ответ на запрос:

#SAV,<HW_Resource>,<Value>

Параметры:

HW_Resource – Тип аппаратного ресурса. Возможные значения:
REL – электромагнтыне реле Rele1 – Rele4
OUT – силовые выходные линии OUT_1 – OUT_5
GIO – слаботочные "выходные" линии IO1 – IO8
PWM – каналы ШИМ PWM_1 – PWM_4

Value – Состояние режима. *1* – включено, *0* – выключено.

Пример 1:

Получить текущее состояние режима сохранения для выходных силовых линий OUT:

запрос: $KE,SAV,OUT,GET
ответ: #SAV,OUT,1

Синтаксис 3: $KE,SAV,CLN

Удалить из энергонезависимой памяти все ранее сохраненные значения всех аппаратных ресурсов.

Ответ на запрос:

#SAV,CLN,OK

Синтаксис 4: $KE,SAV,PER,SET,<Value>

Управление периодом сохранения изменений в энергонезависимую память. Для того чтобы продлить ресурс работы энергонезависимой памяти, изменения сохраняются не мгновенно, а с некоторой задержкой. По умолчанию, изменения будут сохраняться не чаще чем раз в 10 секунд. Настройка сохраняется в энергонезависимой памяти модуля.

Параметры:

Value – Период сохранения в секундах. Целое число от 0 до 255. Значение по умолчанию – 10. Значение 0 – изменения сохраняться не будут.

Ответ на запрос:

#SAV,PER,SET,OK

Синтаксис 5: $KE,SAV,PER,GET

Возвращает текущее значение периода записи изменений состояний ресурсов в память.

Ответ на запрос:

#SAV,PER,<Value>

$KE,WEB

Управление режимом отображения Web интерфейса модуля.

Синтаксис 1: **$KE,WEB,SRC,SET,<Value>**

Определет какой Web интерфейс использовать по умолчанию – встроенный штатный заводской (нередактируемый) или пользовательский редактируемый. Необходима перезагрузка (программный ресет или по питанию) модуля для того что бы изменения вступили в силу.

Заводской интерфейс встроен в образ “прошивки”, не редактируется, хранится во Flash памяти микроконтроллера и перезаписывается только вместе с образом прошивки. Пользовательский интерфейс храниться во внешней Flash памяти модуля, может быть модифицирован и перезаписан.

Параметры:

Value – 0 – заводской (штатный) интерфейс, 1 - пользовательский.

Ответ на запрос:

#WEB,SRC,SET,OK

Пример 1:

Включить использование пользовательского Web интерфейса:

запрос: $KE,WEB,SRC,SET,1
ответ: #WEB,SRC,SET,OK

Синтаксис 2: $KE,WEB,SRC,GET

Возвращает текущее состояние настройки – тип Web интерфейса по умолчанию (заводской / пользовательский).

Ответ на запрос:

#WEB,SRC,<Value>

Синтаксис 3: $KE,WEB,RFR

Инициирует повторную передачу всех данных и редко изменяемых настроек в Web интерфейс. Равнозначно команде Refresh (F5) в Web браузере.

Ответ на запрос:

#WEB,RFR,OK

Сброс настроек

Команды ресета модуля и стирания настроек (возврат к заводским установкам).

$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) включая сетевые настройки.

Синтаксис: $KE,DEFAULT

KE Ke - сообщения

Информационные сообщения, передаваемые модулем через указанные интерфейсы при возникновении событий или по расписанию (с заданным темпом).

Синтаксис

Общий синтаксис Ke-кообщений модуля Laurent-5 иммет вид:

```
#M,<MsgName>,<Parameter_1>,…,<Parameter_N>
```

Параметры:

MsgName – имя Ke-сообщения, например “RELE”

Parameter 1-N – Параметры (информационные поля) конкретного Ke-сообщения.

Ke-сообщения разделяются на две гурппы:

ON_EVENT – Сообщения “По событию”. Выдаются в порт при возникновении определенных событий

ON_TIME – Сообщения “По времени”. Выдаются автоматически с заданной частотой (по умолчанию – 1 Гц).

Список Ke-сообщений:

Имя	Тип	Описание
ECAT	ON_EVENT	Сообщение содержит информацию о произошедшем событии CAT
EIN	ON_EVENT	Выдается при изменении уровня сигнала на входной оптоизолированной линии IN1 - IN6
EIOI	ON_EVENT	Выдается по событию изменения уровня сигнала на линии общего назначения IO1 - IO8 настроенной на вход
RFID	ON_EVENT	Выдается при обнаружении ключа RFID. Содержит информацию об идентификаторе обнаруженного ключа
IBUT	ONEVENT	Выдается при обнаружении ключа iButton (Touch Memory).

		Содержит информацию о номере обнаруженного ключа
DS18	ONEVENT	Результат чтения 'сырых' измерений датчика температуры 1-Wire DS18B20, 9 байт данных включая CRC. Выдается автоматически при очередном считывании показаний.
TIME	ON_TIME	Время (с момента старта и абсолютное из RTC)
RELE	ON_TIME	Состояния реле
IN	ON_TIME	Состояния входных оптоизолированных линий IN
IOD	ON_TIME	Направления линий общего назначения IO
IOI	ON_TIME	Состояния линий IO настроенных на вход
IOO	ON_TIME	Состояния линий IO настроенных на выход
OUT	ON_TIME	Состояния выходных силовых линий OUT
ADCR	ON_TIME	'Сырые' измерения АЦП - цифровой код
ADCV	ON_TIME	Измерения АЦП преобразованные в Вольты
PWM	ON_TIME	Состояния каналов ШИМ
1WT	ON_TIME	Показания датчиков температуры 1-Wire DS18B20
HMD	ON_TIME	Состояние и измерения датчика влажности DHT11
IPLL	ON_TIME	Счетчик импульсов для входных оптоизолированных линий IN
IPLI	ON_TIME	Счетчик импульсов для линий общего назначения IO настроенных на вход
ACS	ON_TIME	Измерения датчиков тока

ECAT

Данное сообщение генерируется автоматически при возникновении событий системы CAT.

Тип: **ONEVENT**

Формат: **#M,ECAT,<CAT_Type>,<Value_1>,<Value_2>**

Параметры:

CAT_Type – Символ, индицирующий тип CAT события. См. Таблицу 1 ниже.

Value 1 – Идентификатор CAT события для которого произошло событие. Целое число в диапазоне [1 - 50]

Value 2 – Данное поле может иметь два различных наполнения (смысла). Если поле числовое – это текущее значение счетчика срабатываний данного события CAT (событие уже произошло). Так же поле может принимать текстовое значение. Возможные варинты:

‘GREEN’ – показания датчика (или другой характерной для данного CAT события величины) вернулись обратно в разрешенную (“зеленую”) зону.

Таблица 1. Типы CAT событий:

Символ	Описание
A	превышение порогов для измерений АЦП
B	обнаружение ключа iButton (Touch Memroy)
C	превышение порогов по числу срабатываний конкретного события CAT
D	превышение порогов для измерений температуры датчика влажности и температуры DHT-11
E	превышение порогов счетчиков импульсов на IO линиях общего назначения настроенных на вход по числу срабатываний
F	Превышение порога для датчика тока
H	превышение порогов для измерений влажности датчика влажности и температуры DHT-11
I	изменение уровня сигнала на дискретной линии общего назначения (IO), настроенной на вход
J	отсутствие изменений уровня сигнала на линиях общего назначения (IO) настроенных на вход в течение заданного времени
K	отсутствие изменений уровня сигнала на входных оптоизолированных линиях (IN) в течение заданного времени
L	изменение уровня сигнала на входной оптоизолированной линии (IN)

M	превышение порогов счетчиков импульсов на входных оптоизолированных линиях IN по числу срабатываний
N	событие по расписанию системного времени
P	выполенение операции PING IP
R	отсутствие активности (прием данных) по порту RS-232 в течение заданного времени
S	событие по расписанию времени RTC (часы реального времени)
T	превышение порогов измерений температуры 1-Wire датчиков температуры DS18B20
W	обнаружение внешним считывателем RFID карты / метки RFID по протоколоу Wiegand
X	превышение порогов по числу переданных (Tx) байт порта RS-232
Y	превышение порогов по числу принятых (Rx) байт порта RS-232

EIN

Данное сообщение генерируется автоматически при изменении уровня сигнала (с низкого на высокий и наоборот) на входной отпоизолированной входной линии IN1 – IN6.

Тип: **ONEVENT**

Формат: **#M,EIN,<Line>,<Value>**

Параметры:

Line	–	Номер входной дискретной линии. Может быть в пределах от 1 до 6 включительно. См. выводы IN1 – IN6.
Value	–	Текущий уровень сигнала на линии: 0 – на линии низкий уровень сигнала 1 – высокий

Пример:

В порт поступило следующее сообщение:

сообщение: #M,EIN,2,1

На входной оптоизолированной линии IN_2 появился сигнал высокого уровня.

EIOI

Данное сообщение генерируется автоматически при изменении уровня сигнала (с низкого на высокий и наоборот) на дискретной линии общего назначения IO1 – IO8 настроенной на вход.

Тип: **ONEVENT**

Формат: **#M,EIOI,<Line>,<Value>**

Параметры:

Line	–	Номер IO линии настроенной на вход. Может быть в пределах от 1 до 8 включительно. См. выводы IO1 – IO8.
Value	–	Текущий уровень сигнала на линии: 0 – на линии низкий уровень сигнала 1 – высокий

Пример:

В порт поступило следующее сообщение:

сообщение: #M,EIOI,8,0

На линии общего назначения (настроенной на вход) IO_8 произошло изменение уровня сигнала. Текущий уровень – низкий (нет напряжения / сигнала).

RFID

Данное сообщение генерируется автоматически при обнаружении внешним считытвателем RFID карты / метки по протоколу Wiegand-26. Сообщение содержит 3-х байтный идентификатор метки.

Тип: **ONEVENT**

Формат: **#M,RFID,<ID_HEX>**

Параметры:

ID_HEX – Идентификатор обнаруженной карты / метки в HEX виде (3 байта, формат передачи данных Wiegand)

Пример:

В порт поступило следующее сообщение:

сообщение: #M,RFID,40A3C7

По интерфейсу Wiegand-26 от внешнего считывателя поступила информация об обнаружении RFID карты / метки с идетификатором 40 A3 C7 (в HEX виде) или 4236231 в десятичном виде.

IBUT

Данное сообщение генерируется автоматически при обнаружении iButton (Touch Memory) метки на шине 1-Wire. Сообщение содержит 8-ми байтный идентификатор метки (адрес устройства 1-Wire).

Тип: **ONEVENT**

Формат: **#M,IBUT,<ID_DEC_DOT>**

Параметры:

ID_DEC_DOT – Идентификатор (адрес 1-Wire) обнаруженной метки iButton в виде строки байт в десятичном виде разделенными точками

Пример:

В порт поступило следующее сообщение:

сообщение: #M,IBUT,1.232.63.79.1.0.0.143

На шине 1-Wire обнаружена метка iButton (Touch Memory) DS1990 с идентификатором 1.232.63.79.1.0.0.143

DS18

Сообщение содержит результат чтения 'сырых' измерений датчика температуры 1-Wire DS18B20 в виде 9 байт данных включая CRC (см. DataSheet датчика температуры DS18B20). Генерируется автоматически при периодическом считывания показаний.

Тип: **ONEVENT**

Формат: **#M,DS18,<Sensor ID>,<Data Packet>**

Параметры:

Sensor ID	–	Номер датчика DS18B20 (целое число [1 - 30]) обнаруженного на шине и опрашиваемого модулем
Data Packet	–	Результат считывания показаний датчика. Пакет из 9 байт в HEX виде

Пример:

В порт поступило следующее сообщение:

сообщение: #M,DS18,1,A1014B467FFF0F10D9

На шине 1-Wire присутствует датчик DS18B20. Во внутреннем списке модуля обнаруженных датчиков, он имеет номер 1. Результат считывания его показаний (9 байт):
A1 01 4B 46 7F FF 0F 10 D9

TIME

Информация о времени: время с момента страта модуля в секундах и абсолютное время от часов реального времени (RTC) включая год, месяц, день месяца, день недели, часы, минуты, секунды. Генерируется с заданным фиксированным темпом (по умолчанию – 1 Гц).

Тип: **ONTIME**

Формат:

#M,TIME,<UpTime>,<RTC_Y>,<RTC_M>,<RTC_D>,<RTC_DW>,<RTC_h>,<RTC_m>,<RTC_s>

Параметры:

UpTime – Время в секундах с момента страта модуля, целое число в диапазоне 0 - 4294967295

RTC_Y – Текущий год (RTC)

RTC_M – Текущий номер месяца, целое число [1 - 12]

RTC_D – Номер дня в течение месяца, целое число [1- 31]

RTC_WD – Номер дня в течение недели, целое число [1 - 7]

RTC_h – Номер часа в течение дня, целое число [0 - 23]

RTC_m – Минуты, целое число [0 - 59]

RTC_s – Секунды, целое число [0 - 59]

Пример:

В порт поступило следующее сообщение:

сообщение: #M,TIME,6235,2019,8,31,6,15,17,30

В данный момент времени модуль отработал 6235 сек с момента старта (последнего ресета). Текущее абсолютное время RTC: 31 Августа 2019, 15 часов, 17 минут, 30 сек.

RELE

Сообщение содержит сводную информацию о текущем состоянии всех реле модуля (включено / выключено). Генерируется с заданным фиксированным темпом (по умолчанию – 1 Гц).

Тип: **ONTIME**

Формат: **#M,RELE,<Value>**

Параметры:

Value – Сводная строка данных содержащая информацию о всех реле модуля. Нумерация в строке производится слева на право. Первому символу в строке соответствует 1-ое реле, второму символу 2-ое реле и т.д.
0 – реле выключено
1 – включено

Пример:

В порт поступило следующее сообщение:

сообщение: #M,RELE,0010

В данный момент времени все реле модуля выключены, кроме 3-го – оно включено.

IN

Сообщение содержит сводную информацию о текущем состоянии всех входных оптоизолированных линий IN1 – IN6 (есть сигнал / напряжение или нет). Генерируется с заданным фиксированным темпом (по умолчанию – 1 Гц).

Тип: **ONTIME**

Формат: **#M,IN,<Value>**

Параметры:

Value – Сводная строка данных содержащая информацию о всех входных оптоизолированных линиях IN. Нумерация в строке производится слева на право. Первому символу в строке соответствует линия IN_1, второму символу IN_2 и т.д.
0 – сигнала (напряжения) на входе нет
1 – есть

Пример:

В порт поступило следующее сообщение:

сообщение: #M,IN,011111

В данный момент времени на всех входных отпоизолированных линиях присутствует сигнал (напряжение), кроме линии IN_1 – на ней сигнала (напряжения) нет.

IOD

Сообщение содержит сводную информацию о текущей настройке (вход или выход) линий общего назначения IO1 – IO8. Генерируется с заданным фиксированным темпом (по умолчанию – 1 Гц).

Тип: **ONTIME**

Формат: **#M,IOD,<Value>**

Параметры:

Value – Сводная строка данных содержащая информацию о настройке направления (вход / выход) всех линий общего назначения IO. Нумерация в строке производится слева на право. Первому символу в строке соответствует линия IO_1, второму символу IO_2 и т.д.
0 – линия настроена на выход
1 – на вход

Пример:

В порт поступило следующее сообщение:

сообщение: #M,IOD,01000100

В данный момент времени линии IO_2 и IO_6 настроены на вход, остальные линии – на выход.

IOI

Сообщение содержит сводную информацию о текущем состоянии (есть входной сигнал / нет) линий общего назначения IO1 – IO8 настроенных на вход. Генерируется с заданным фиксированным темпом (по умолчанию – 1 Гц).

Тип: **ONTIME**

Формат: **#M,IOI,<Value>**

Параметры:

Value – Сводная строка данных содержащая информацию о состоянии (есть входной сигнал / нет) всех линий общего назначения IO. Нумерация в строке производится слева на право. Первому символу в строке соответствует линия IO_1, второму символу IO_2 и т.д.
0 – на входной линии сигнала нет
1 – есть сигнал (напряжение)
x – линия настроена на выход

Пример:

В порт поступило следующее сообщение:

сообщение: #M,IOI,1000000x

В данный момент времени на входной линии IO_1 есть входной сигнал (высокий логический уровень), на линиях IO_2 – IO_7 сигнала нет, а линия IO_8 настроена на выход.

IOO

Сообщение содержит сводную информацию о текущем состоянии (включена / выключена) линий общего назначения IO1 – IO8 настроенных на выход. Генерируется с заданным фиксированным темпом (по умолчанию – 1 Гц).

Тип: **ONTIME**

Формат: **#M,IOO,<Value>**

Параметры:

Value – Сводная строка данных содержащая информацию о состоянии (включена / выключена) всех линий общего назначения IO настроенных на выход. Нумерация в строке производится слева на право. Первому символу в строке соответствует линия IO_1, второму символу IO_2 и т.д.
0 – линия выключена
1 – включена
x – линия настроена на вход

Пример:

В порт поступило следующее сообщение:

сообщение: #M,IOO,11111001

В данный момент времени все линии IO настроена на выход, все линии включены кроме IO_6 и IO_7 (они выключены).

OUT

Сообщение содержит сводную информацию о текущем состоянии (включена / выключена) выходных силовых линий OUT. Генерируется с заданным фиксированным темпом (по умолчанию – 1 Гц).

Тип: **ONTIME**

Формат: **#M,OUT,<Value>**

Параметры:

Value – Сводная строка данных содержащая информацию о состоянии (включена / выключена) всех силовых линий OUT. Нумерация в строке производится слева на право. Первому символу в строке соответствует линия OUT_1, второму символу OUT_2 и т.д.
0 – линия выключена
1 – включена

Пример:

В порт поступило следующее сообщение:

сообщение: #M,OUT,11100

В данный момент времени линии OUT_1 – OUT_3 включены, линии OUT_4 – OUT_5 выключены.

ADCR

Сообщение содержит результаты АЦП преобразования в виде “сырого” 10-ти битного цифрового кода в диапазоне значений [0 - 1023] для всех каналов АЦП. Генерируется с заданным фиксированным темпом (по умолчанию – 1 Гц).

Тип: **ONTIME**

Формат: **#M,ADCR,<Value_1>,…,<Value_5>**

Параметры:

Value 1-5 – Измерения АЦП в виде “сырого” цифрового кода [0-1023] для всех 5-ти каналов АЦП модуля (ADC1 – ADC5).

Пример:

В порт поступило следующее сообщение:

сообщение: #M,ADCR,0,1023,678,0,0

В текущий момент времени на каналах АЦП 1,4,5 входной сигнал отсутствует. Каналы АЦП 2 и 3 выполнили измерения входного сигнала и “сырые” цифровые коды эти измерений равны 1023 и 678 соответственно.

ADCV

Сообщение содержит результаты преобразования всех каналов АЦП в виде значения напряжения (в Вольтах) пересчитанного из “сырого” цифрового кода с учетом значения напряжения источника опорного напряжения, аппартных и пользовательских коэффициентов делителя напряжения. Генерируется с заданным фиксированным темпом (по умолчанию – 1 Гц).

Тип: **ONTIME**

Формат: **#M,ADCV,<Value_1>,…,<Value_5>**

Параметры:

Value 1-5 – Измерения АЦП в Вольтах для всех 5-ти каналов АЦП модуля (ADC1 – ADC5).

Пример:

В порт поступило следующее сообщение:

сообщение: #M,ADCV,0,2.5,0,0,1.45

В текущий момент времени на каналах АЦП 1,3,4 входной сигнал отсутствует (напряжение – ноль). Каналы АЦП 2 и 5 выполнили измерения напряжения входного сигнала (с учетом пользовательских и аппаратных коэффициентов делителей напряжения) и эти измерений равны 2.5 и 1.45 В соответственно.

PWM

Сообщение содержит текущие значения уровней мощности выходного ШИМ сигнала в % для всех каналов ШИМ модуля. Генерируется с заданным фиксированным темпом (по умолчанию – 1 Гц).

Тип: **ONTIME**

Формат: **#M,PWM,<Value_1>,…,<Value_4>**

Параметры:

Value 1-5 – Уровень мощности ШИМ сигнала в % для всех 4 х каналов ШИМ модуля (PWM1 – PWM4).

Пример:

В порт поступило следующее сообщение:

сообщение: #M,PWM,100,0,80,0

В текущий момент времени каналы ШИМ 2 и 4 выключены (нулевой уровень мощности сигнала). Каналы 1 и 3 включены и на них выставлены уровни в 100% и 80% соответственно.

1WT

Сообщение содержит показания 1-Wire датчика температуры (класса DS18B20) в градусах Цельсия и 1-Wire ID (идентификатор). Генерируется с заданным фиксированным темпом (по умолчанию – 1 Гц). Сообщение генерируется отдельно для каждого из обнаруженных датчиков температуры.

Тип: **ONTIME**

Формат: **#M,1WT,<HEX_ID,<Value>**

Параметры:

HEX_ID – Уникальный идентификатор датчика температуры 1-Wire в HEX виде

Value – Показания датчика температуры в градусах Цельсия

Пример:

В порт поступило следующее сообщение:

сообщение: #M,1WT,28091FEA09000047,26.06

Измерения температуры 1-Wire датчика с ID в HEX 28091FEA09000047 (или в десятичном виде 40.9.31.234.9.0.0.71) равны +26.06 ºC

HMD

Сообщение содержит текущий статус датчика влажности и температуры класса DHT-11 (подключен / неподключен / данные валидны) и измерения датчика (влажность, температура). Генерируется с заданным фиксированным темпом (по умолчанию – 1 Гц).

Тип: ONTIME

Формат: #M,HMD,<Connected>,<Valid>,<Humidity>,<Temperature>

Параметры:

Connected	–	Состояние датчика: 0 – датчик не подключен (не обнаружен) 1 – датчик подключен
Valid	–	Флаг валидности измерений: 0 – измерения не валидны (использовать нельзя) 1 – измерения валидны
Humidity	–	Относительная влажность в %
Temperature	–	Температура в градусах Цельсия

Пример:

В порт поступило следующее сообщение:

сообщение: #M,HMD,1,1,35.0,26.0

В текущий момент датчик обнаружен, измерения корректные, влажность 35%, температура +26 C°.

IPLL

Сообщение содержит показания счетчиков импульсов ассоциированных с входными оптоизолированными линиями IN1 – IN6. Генерируется с заданным фиксированным темпом (по умолчанию – 1 Гц).

Тип: ONTIME

Формат: #M,IPLL,<Value_1>,…,<Value_6>

Параметры:

Value 1-6 – Показания счетчиков импульсов для всех входных оптоизолированных линий IN1 – IN6

Пример:

В порт поступило следующее сообщение:

сообщение: #M,IPLL,0,0,0,0,0,200

Показания всех счетчиков импульсов для входных оптоизолированных линий IN1 – IN6 равны нулю кроме линии IN6 – для которой показания равны 200 импульсов.

ACS

Сообщение содержит измерения датчиков тока подключенных к каналам АЦП ADC_2 – ADC_5. Генерируется с заданным фиксированным темпом (по умолчанию – 1 Гц).

Тип: ONTIME

Формат: #M,ACS,<Value_2>,…,<Value_5>

Параметры:

Value 2-5 – Показания датчиков тока в Амперах для четырех датчиков тока. Если датчик не подключен – в соответствующем поле выдается значение 0.

Пример:

В порт поступило следующее сообщение:

сообщение: #M,ACS,18.111,0.005,0.000,12.927

На текущий момент времени показания датчиков тока (подключенных к соответствующим каналам АЦП) следующее:

ADC_2: 18.111 A
ADC_3: 0.005 A
ADC_4: 0.000 A (либо датчик неподключен)
ADC_5: 12.927 A

DynVAR Динамические переменные

В составе некоторых команд и сообщений можно получить / передать не только заранее заданный статический текст, но и текущие динамические значения и показания датчиков и системных параметров, подставляемых в строку в режиме реального времени.

Одной из типовых задач является отпрвка SMS сообщения при возникновении события (например, тревога по датчику) с текущим показанием датчика и актуальным временем на момент возникновения события.

Для решения этой и аналогичхных задач в модулях Laurent-5 / 5G предназначены т.н. *Динамические переменные*. Это набор определенных текстовых полей в специальном формате которые заменяются модулем динамически на текущее значение параметра (например, показания датчика) и в таком финальном виде передаются / выдаются в виде KE-сообщения, параметров URL команды или текста SMS сообщения.

Синтаксис

Общий синтаксис динамических переменных модуля Laurent-5 / 5G иммет вид:

```
~<Name><LineId>~
```

Состоит из следующих друг за другом полей <Name>, опционального поля <LineID> и ограничены символом “тильда” (~) с обоих сторон.

Параметры:

Name – имя динамической перменной, длина 2 символа. Например, имя динамической переменной для состояний электромагнитных реле имеет вид *‘RL’*

LineId – Опциональный параметр. Целое число до 0 до 99 определяющее номер аппаратного ресурса (если применем к данному типу переменной). Например, если нам нужно получить состояние реле RELE_2 то следует использовать переменную *~RL2~*. Допустима запись *~RL2~* как ровно и *~RL02~*

Список переменных

В качестве примеров ниже, динамические перемнные используются в составе данных передаваемых:

1) в порт TCP сервера с использованием Ke-команды $KE,PUT
2) через HTTP GET запрос на внешний сервер (192.168.0.200, 80 порт) с скрпиту scr.cgi

Возможно применение динамических переменных и в составе других Ke-команд.

Название:	LF
Описание:	Символ переноса строки (\r или 0x0A в HEX). Может быть полезен при составлении SMS сообщения, что бы разместить отдельные данные на разных строках.
Параметры (LineID):	нет
Пример:	~LF~
Использование:	`$KE,PUT,S,C,Line1~LF~Line2~LF~Line3`

Название:	DT
Описание:	Дата (год, меясц, день) из RTC
Параметры (LineID):	нет
Пример:	~DT~
Использование:	`$KE,PUT,S,C,Date: ~DT~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~DT~`

Название:	TM
Описание:	Время (час, минута, секунда) из RTC
Параметры (LineID):	нет
Пример:	~TM~
Использование:	`$KE,PUT,S,C,Time: ~TM~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~TM~`

Название:	TS
Описание:	Время с момента страта модуля в сеукндах
Параметры (LineID):	нет
Пример:	~TS~
Использование:	`$KE,PUT,S,C,UpTime: ~TS~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~TS~`

Название:	RL
Описание:	Состояние реле (включено / выключено)
Параметры (LineID):	Номер реле [1 - 4]
Пример:	~RL2~
Использование:	`$KE,PUT,S,C,RELE ~RL2~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~RL2~`

Название:	IN
Описание:	Состояние входной оптоизолированной линии IN1 – IN6 (1 - есть / 0 - нет сигнала на входе)
Параметры (LineID):	Номер входной оптоизолированной линии IN1-IN6
Пример:	~IN6~
Использование:	`$KE,PUT,S,C,~IN1~ ~IN2~ ~IN3~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~IN3~`

Название:	OT
Описание:	Состояние выходной силовой линии OUT1 – OUT5 (1 - включена / 0 - выключена)
Параметры (LineID):	Номер выходной силовой линии OUT1 – OUT5
Пример:	~OT3~
Использование:	`$KE,PUT,S,C,OUT: ~OT4~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~OT4~`

Название:	GO
Описание:	Состояние линии IO настроенной на выход (1 - включена / 0 - выключена)
Параметры (LineID):	Номер дискретной двунаправленной линии общего назначения IO1 – IO8
Пример:	~GO7~
Использование:	`$KE,PUT,S,C,LedState: ~GO7~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~GO7~`

Название:	GI
Описание:	Состояние линии IO настроенной на вход (1 - есть / 0 - нет сигнала на входе)
Параметры (LineID):	Номер дискретной двунаправленной линии общего назначения IO1 – IO8
Пример:	~GI6~
Использование:	`$KE,PUT,S,C,Button: ~GI6~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~GI6~`

Название:	GD
Описание:	Состояние настройки линии IO (1 - вход или 0 - выход)
Параметры (LineID):	Номер дискретной двунаправленной линии общего назначения IO1 – IO8
Пример:	~GD3~
Использование:	`$KE,PUT,S,C,~GD3~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~GD3~`

Название:	AV
Описание:	Напряжение в Вольтах на входе канала АЦП ADC1 – ADC5
Параметры (LineID):	Номер канала АЦП ADC1 – ADC5
Пример:	~AV2~
Использование:	`$KE,PUT,S,C,Voltage: ~AV2~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~AV2~`

Название:	PW
Описание:	Значение уровня мощности ШИМ сигнала
Параметры (LineID):	Номер ШИМ канала [1 - 4]
Пример:	~PW1~
Использование:	`$KE,PUT,S,C,PWM: ~PW1~,~PW2~,~PW3~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~PW1~`

Название:	1T
Описание:	Температура в градусах Цельсия датчика температуры 1-Wire DS18B20
Параметры (LineID):	Номер датчика температуры 1-Wire класса DS18B20, [1 - 20]
Пример:	~1T15~
Использование:	`$KE,PUT,S,C,Temperature: ~1T15~ C` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~1T15~`

Название:	Dh
Описание:	Показания влажности цифрового датчика влажности и температуры класса DHT-11 / 22
Параметры (LineID):	Номер датчика влажности и температуры класса DHT-11 / 22
Пример:	~Dh1~
Использование:	`$KE,PUT,S,C,HUM: ~Dh1~%` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~Dh1~`

Название:	Dt
Описание:	Показания температуры цифрового датчика влажности и температуры класса DHT-11 / 22
Параметры (LineID):	Номер датчика влажности и температуры класса DHT-11 / 22
Пример:	~Dt1~
Использование:	`$KE,PUT,S,C,TMP: ~Dt1~C` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~Dt1~`

Название:	IC
Описание:	Телефонный номер входящего звонка (последний зафиксированный). Применим только для модуля Laurent-5G (с GSM)
Параметры (LineID):	нет
Пример:	~IC~
Использование:	`$KE,PUT,S,C,LastCall: ~IC~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~IC~`

Название:	IS
Описание:	Телефонный номер входящей SMS (последний зафиксированный). Применим только для модуля Laurent-5G (с GSM)
Параметры (LineID):	нет
Пример:	~IS~
Использование:	`$KE,PUT,S,C,LastSMS Phone: ~IS~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~IS~`

Название:	UV
Описание:	Значение пользовательской переменной. См. команду $KE,VAR
Параметры (LineID):	Номер пользовательской переменной
Пример:	~UV1~
Использование:	`$KE,PUT,S,C,MyVAR_1: ~UV1~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~UV1~`

Название:	WG
Описание:	ID последней обнаруженной RFID метки по протоколу Wiegand
Параметры (LineID):	нет
Пример:	~WG~
Использование:	`$KE,PUT,S,C,LastRFID: ~WG~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~WG~`

Название:	IB
Описание:	ID последней обнаруженной iButton метки (DS1990) по протоколу 1-Wire
Параметры (LineID):	нет
Пример:	~IB~
Использование:	`$KE,PUT,S,C,LastIButton: ~IB~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~IB~`

Название:	IL
Описание:	Счетчик импульсов для входных оптоизолированных линии IN1 – IN6
Параметры (LineID):	Номер входной оптоизолированной линии IN1-IN6
Пример:	~IL2~
Использование:	`$KE,PUT,S,C,~IL2~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~IL3~`

Название:	II
Описание:	Счетчик импульсов для дискретных линий общего назначения IO1 – IO8 настроенных “на вход”
Параметры (LineID):	Номер дискретной двунаправленной линии общего назначения IO1 – IO8
Пример:	~II8~
Использование:	`$KE,PUT,S,C,~II8~` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~II8~`

Название:	AC
Описание:	Показания датчика тока в Амперах, подключенного ко входам АЦП ADC2 – ADC5
Параметры (LineID):	Номер канала АЦП ADC2 – ADC5. Возможные значения [2-5]
Пример:	~AC3~
Использование:	`$KE,PUT,S,C,Current: ~AC3~ A` `$KE,URL,RUN,192.168.0.200,80,scr.cgi?val=~AC3~`

Примеры

Например, если в случае тревоги (например, как реакция на CAT событие), выполнить Ke-команду (отправка данных в порт TCP серевра):

```
$KE,PUT,S,C,TREVOGA!~LF~H = ~Dh1~%~LF~IN_2: ~IN2~~LF~IN_3: ~IN3~~LF~~DT~ ~TM~
```

Или аналогично в виде SMS сообщения:

```
$KE,SMS,SND,1,C,TREVOGA!~LF~H = ~Dh1~%~LF~IN_2: ~IN2~~LF~IN_3: ~IN3~~LF~~DT~ ~TM~
```

То можно передать и получить не просто “сухой” текст о том, что произошла тревожная ситуация а так же увидеть показания некоторых конкретных датчиков (в данном случае влажность датчика DHT-11 и состояние входных оптоизолированных линий IN_2 и IN_3) и актуальную дату и время из RTC на момент возникновения события:

KernelChip
Interface solutions

© *2012 - 2021 KERNELCHIP* Компоненты и модули для управления, мониторинга и автоматизации

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