

Ke-USB24R

Команды управления модулем

Версия 1.04

13 декабря 2009

История документа:

Версия	Дата	Описание
1.04	13-12-2009	Добавлено описание команд: \$KE,FW \$KE,WRA \$KE,RID \$KE,RDR
1.03	15-11-2009	Добавлено описание команды \$KE,AFR Дополнено описание команды \$KE,ADC
1.02	17-07-2009	Первая версия документа

Содержание

Введение.....	4
Команда \$KE.....	5
Команда \$KE,FW.....	5
Команда \$KE,WR.....	5
Команда \$KE,WRA.....	6
Команда \$KE,RD.....	7
Команда \$KE,RID.....	8
Команда \$KE,REL.....	10
Команда \$KE,RDR.....	10
Команда \$KE,AFR.....	11
Команда \$KE,ADC.....	12
Группа Команд \$KE,IO.....	13
Команда \$KE,IO,SET.....	13
Команда \$KE,IO,GET.....	14
Группа Команд \$KE,UD.....	15
Команда \$KE,UD,SET.....	15
Команда \$KE,UD,GET.....	16
Группа Команд \$KE,USB.....	16
Команда \$KE,USB,SET.....	16
Команда \$KE,USB,GET.....	17
Команда \$KE,SER.....	17
Команда \$KE,RST.....	17

Введение

Для управления модулем KE-USB24R предназначен ряд команд в текстовом формате, называемых KE командами. 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).

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>.

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

Команда \$KE

Синтаксис: \$KE

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

Ответ на запрос:

#OK

Пример:

Тестовая проверка модуля:

запрос: \$KE
ответ: #OK

Команда \$KE,FW



Синтаксис: \$KE,FW

Команда возвращает номер версии программного обеспечения модуля.

Ответ на запрос:

#FW,<Version>

Ограничения:

Данная команда доступна начиная со 2-ой версии модуля Ke-USB24R. Модули предыдущих версий возвращают сообщение об ошибке при ее запросе.

Пример:

Запрос версии программного обеспечения модуля:

запрос: \$KE,FW
ответ: #FW,2.0

Команда \$KE,WR

Синтаксис: \$KE,WR,<LineNumber>,<Value>

С помощью данной команды можно установить высокий ($Value = 1$) или низкий уровень напряжения ($Value = 0$) на линии ввода/вывода модуля под номером *LineNumber*. Линия *LineNumber* должна быть настроена на выход.

Параметры:

LineNumber – номер линии ввода/вывода. Может быть в пределах от 1 до 18 включительно.

Value – значение для записи на линию. 1 – высокий уровень напряжения, 0 – низкий уровень напряжения.

Ответ на запрос:

#WR,OK – значение успешно установлено.

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

Пример:

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

запрос: \$KE,WR,16,1

ответ: #WR,OK

Команда \$KE,WRA



Синтаксис: \$KE,WRA,<ArrayOfValues>

Команда \$KE,WRA позволяет за одно обращение установить высокий или низкий уровень напряжения на всех линиях модуля, настроенных на выход.

Параметры:

ArrayOfValues – строка длиной от 1 до 18 символов. Может содержать символы '0' или '1'. Нумерация символов в строке производится слева на право. Значение первого символа строки будет установлено на линии вывода номер 1, значение второго символа - на линии номер 2 и т.д. Если соответствующая линия настроена на вход, то операция записи производится не будет. Строка может содержать меньшее число символов, чем суммарное число линий модуля, например, строка из 4-х символов позволит установить первые четыре линии модуля.

Ответ на запрос:

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

Ограничения:

Данная команда доступна начиная со 2-ой версии модуля Ke-USB24R. Модули предыдущих версий возвращают сообщение об ошибке при ее запросе.

Пример 1:

Считаем, что все линии модуля настроены на выход. Установим на всех линиях логическую единицу:

запрос: \$KE,WRA,11111111111111111111
 ответ: #WRA,OK,18

Пример 2:

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

запрос: \$KE,WRA,11111111111111111111
 ответ: #WRA,OK,15

Входные линии действием этой команды затронуты не будут. При этом значения на всех выходных линиях будут обновлены. Как видно, ответ содержит информацию о том, что были успешно обновлены только 15 линий.

Пример 3:

Считаем, что все линии модуля настроены на выход. Установим на первых 8-ми линиях модуля логический ноль:

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

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

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

Параметры:

LineNumber – номер линии ввода/вывода. Может быть в пределах от 1 до 18 включительно.

Ответ на запрос:

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

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

Пример:

Считать информацию с линии 2:

запрос: \$KE,RD,2
 ответ: #RD,02,1

Синтаксис (Вариант 2): \$KE,RD,ALL

По данной команде модуль произведет последовательный перебор всех линий ввода/вывода. Если линия настроена на вход будет произведено считывание информации с этой линии. Если линия настроена на выход, считывание производиться не будет. Результат выводится в виде сводной строки данных, состоящей из 18 символов. Нумерация позиции символа в строке осуществляется слева на право и соответствует номеру линии. Если линия под номером N настроена на вход, то символ в сводной строке под номером N будет содержать результат чтения. Если линия настроена на выход – в соответствующей позиции строки будет выставлен символ 'x'.

Ответ на запрос:

```
#RD,<Line1 Value>< Line2 Value>< Line3 Value>.... <Line18 Value>
```

Пример:

Считать информацию со всех линий модуля:

```
запрос: $KE,RD,ALL
ответ: #RD,xxx10xxx0xxx1xxxxx
```

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

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

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

Параметры:

LineNumber – номер линии ввода/вывода. Может быть в пределах от 1 до 18 включительно.

Ответ на запрос:

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

Ограничения:

Данная команда доступна начиная со 2-ой версии модуля Ke-USB24R. Модули предыдущих версий возвращают сообщение об ошибке при ее запросе.

Пример 1:

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

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

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

Пример 2:

Считать информацию с входной линии номер 2:

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

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

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

Параметры:

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

Ответ на запрос:

```
#RID,<Type>,<Line1 Value>< Line2 Value>< Line3 Value>.... <Line18 Value>
```

Ответ за запрос содержит информацию по всем 18 линиям в виде сводной строки данных. Нумерация в строке производится слева на право. Первому символу в строке соответствует линия номер 1, второму символу линия номер 2 и т.д. *Line Value = 0* – на линии установлен низкий логический уровень, *Line Value = 1* – соответственно, высокий логический уровень. Если направление линии не соответствует запрошенному, соответствующий символ в ответной сводной строке будет установлен в значение 'x'.

Пример:

Считать информацию со всех линий модуля:

```
запрос:  $KE,RID,ALL
ответ:   #RID,ALL,000101110011111001
```

запрос: \$KE,RID,IN
 ответ: #RID,IN,xxx10xxx0xxx1xxxxx

запрос: \$KE,RID,OUT
 ответ: #RID,OUT,000xx111x011x11001

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

Команда \$KE,REL

Синтаксис: \$KE,REL,<ReleNumber>,<Value>

Команда предназначена для управления реле модуля.

Параметры:

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

Value – управляющее значение. 0 – контакты реле 1 и 2 замкнуты, 2 и 3 разомкнуты;
 1 – контакты реле 1 и 2 разомкнуты, 2 и 3 замкнуты (реле включено).

Ответ на запрос:

#REL,OK – значение успешно установлено.

Пример:

Замкнем контакты 2 и 3 второго реле:

запрос: \$KE,REL,2,1
 ответ: #REL,OK

Команда \$KE,RDR



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

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

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.

Ограничения:

Данная команда доступна начиная со 2-ой версии модуля Ke-USB24R. Модули предыдущих версий возвращают сообщение об ошибке при ее запросе.

Пример:

Запросим состояние 3-го реле модуля:

запрос: \$KE,RDR,3
ответ: #RDR,3,1

Синтаксис (Вариант 2): \$KE,RDR,ALL

Команда позволяет получить состояние всех четырех реле за один запрос.

Параметры:

Отсутствуют.

Ответ на запрос:

#RID,ALL,<Rele1 State>,<Rele2 State>,<Rele3 State>,<Rele4 State> – запрос состояния реле произведено успешно. *State = 0* – реле выключено, *State = 1* – соответственно, реле включено.

Ограничения:

Данная команда доступна начиная со 2-ой версии модуля Ke-USB24R. Модули предыдущих версий возвращают сообщение об ошибке при ее запросе.

Пример:

Запросим состояние всех реле модуля:

запрос: \$KE,RDR,ALL
ответ: #RDR,ALL,0,1,1,1

Команда \$KE,AFR**Синтаксис: \$KE,AFR,<Frequency>**

Команда устанавливает частоту автоматического опроса каналов АЦП модуля. Модуль Ke-USB24R позволяет настроить автоматическую отправку значений АЦП в порт USB. Для этого необходимо задать частоту опроса для всех каналов с помощью этой команды. Затем нужно разрешить автоматический опрос необходимого канала/каналов АЦП. (см. описание команды \$KE,ADC)

Параметры:

Frequency – частота опроса в герцах. Может принимать значения [0 - 400].

Ответ на запрос:

#AFR,OK – значение успешно установлено.

Пример:

Установим частоту автоматического опроса каналов АЦП равной 150 Гц:

запрос: \$KE,AFR,150

ответ: #AFR,OK

Команда \$KE,ADC

Синтаксис: \$KE,ADC,<ChannelNumber>[,<AutoFlag>]

Считывание результата АЦП с канала под номером *ChannelNumber*.

Параметры:

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

AutoFlag – опциональный параметр. Флаг, разрешающий или запрещающий автоматический опрос данного канала АЦП. Может принимать значения 0 или 1. 1 – автоматический опрос разрешен, 0 – запрещен.

Ответ на запрос:

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

$$V_{in} = \frac{Value}{1023} \cdot 5 [B]$$

Пример 1:

Получить результат преобразования АЦП с 3-го канала:

запрос: \$KE,ADC,3

ответ: #ADC,3,0645

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

$$V_{in} = \frac{645}{1023} \cdot 5 = 3.152 [B]$$

Пример 2:

Включить автоматический опрос 2-го канала АЦП:

запрос: \$KE,ADC,2,1

Группа Команд \$KE,IO

Команды группы IO (Input/Output) предназначены для управления режимом ввода/вывода линий модуля, позволяя настраивать линии на вход или на выход.

Команда \$KE,IO,SET

Синтаксис: \$KE,IO,SET,<LineNumber>,<IoDirection>[,S]

Команда позволяет настроить линию ввода/вывода *LineNumber* в качестве выходной (*IoDirection* = 0) или входной (*IoDirection* = 1). Если установлен необязательный флаг *S* (*Save*), то направление (вход/выход) линии *LineNumber* будет сохранено в энергонезависимой памяти модуля. При последующих включениях модуля линия *LineNumber* будет автоматически сконфигурирована в то направление, которое было сохранено ранее.

Параметры:

LineNumber – номер линии ввода/вывода. Может быть в пределах от 1 до 18 включительно.

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

S – флаг сохранения настройки в энергонезависимой памяти модуля.

Ответ на запрос:

#IO,SET,OK

Пример 1:

Установить линию 5 в качестве выходной, в памяти не сохранять:

запрос: \$KE,IO,SET,5,0

ответ: #IO,SET,OK

Пример 2:

Установить линию 3 в качестве входной, сохранить настройку в памяти:

запрос: \$KE,IO,SET,3,1,S

ответ: #IO,SET,OK

Команда \$KE,IO,GET**Синтаксис (Вариант 1): \$KE,IO,GET,<Location>**

Вывод значений направления ввода/вывода для всех 18 линий модуля. Информация будет соответствовать текущему состоянию, если поле *Location* равно 'CUR' или будет взята из энергонезависимой памяти ('MEM'), т.к. в общем случае значения могут отличаться (если подавалась команда \$KE,IO,SET без указания сохранять в энергонезависимой памяти).

Параметры:

Location – значение этого параметра определяет будет ли информация о направлении линий соответствовать текущему состоянию, или будет прочитана из энергонезависимой памяти.

CUR – текущее состояние

MEM – энергонезависимая память

Ответ на запрос:

#IO, <Line1 Direction><Line2 Direction>...<Line18 Direction>

Line Direction = 1 – линия настроена на вход

Line Direction = 0 – линия настроена на выход

Пример 1:

Получить значения направлений ввода/вывода линий модуля на текущий момент времени:

запрос: \$KE,IO,GET,CUR

ответ: #IO,000100001100000000

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

Пример 2:

Получить значения направлений ввода/вывода линий модуля, сохраненных в энергонезависимой памяти модуля:

запрос: \$KE,IO,GET,MEM

ответ: #IO,100000001000000000

В энергонезависимой памяти модуля для линий с номерами 1, 9 сохранено направление на вход, для остальных – на выход.

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

Вывод значения направления ввода/вывода для линии *LineNumber*. Информация будет соответствовать текущему состоянию, если поле *Location* равно 'CUR' или будет взята из энергонезависимой памяти ('MEM'), т.к. в общем случае значения могут отличаться (если подавалась команда \$KE,IO,SET без указания сохранять в энергонезависимой памяти).

Параметры:

LineNumber – номер линии ввода/вывода. Может быть в пределах от 1 до 18 включительно.

Location – значение этого параметра определяет будет ли информация о направлении линий соответствовать текущему состоянию, или будет прочитана из энергонезависимой памяти.

CUR – текущее состояние

MEM – энергонезависимая память

Ответ на запрос:

#IO,<Value>

Value – направление ввода/вывода, 0 – выход, 1 – вход.

Пример 1:

Получить значение направления ввода/вывода линий модуля 13 на текущий момент времени:

запрос: \$KE,IO,GET,CUR,13

ответ: #IO,1

Пример 2:

Получить значение направления ввода/вывода линий модуля 13, сохраненного в энергонезависимой памяти модуля:

запрос: \$KE,IO,GET,MEM,13

ответ: #IO,0

Группа Команд \$KE,UD

Команды группы UD (User Data) предназначены для управления данными пользователя, которые модуль Ke-USB24R позволяет сохранять в своей энергонезависимой памяти.

Команда \$KE,UD,SET

Синтаксис: \$KE,UD,SET,<Data>

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

Параметры:

Data – данные для записи в память; не более 32 байт (символов).

Ответ на запрос:

#UD,SET,OK

Пример:

Сохранить в энергонезависимой памяти модуля строку 'My Data for storage':

запрос: \$KE,UD,SET,My Data for storage
ответ: #UD,SET,OK

Команда \$KE,UD,GET

Синтаксис: \$KE,UD,GET

Чтение ранее сохраненных пользователем данных из энергонезависимой памяти модуля.

Ответ на запрос:

#UD,<Data> - чтение прошло успешно, поле *Data* содержит прочтенные данные

#UD,NOTSET – данные отсутствуют.

Пример:

Считать данные энергонезависимой памяти модуля:

запрос: \$KE,UD,GET
ответ: #UD, My Data for storage

Группа Команд \$KE,USB

Команды группы USB предназначены для управления строковым дескриптором USB устройства. Информация содержащаяся в строковом дескрипторе устройства отображается в виде сообщения операционной системы при подключении модуля к шине USB компьютера. По умолчанию используется строка 'Ke-USB24R'. Команды этой группы позволяют установить свою собственную строку дескриптора устройства и сохранить ее в памяти модуля.

Команда \$KE,USB,SET

Синтаксис: \$KE,USB,SET,<Data>

Позволяет сохранить строку дескриптора USB.

Параметры:

Data – строка для записи в память; не более 32 байт (символов).

Ответ на запрос:

#USB,SET,OK

Пример:

Установить в качестве строки дескриптора USB устройства текст 'My USB Device':

запрос: \$KE,USB,SET, My USB Device
ответ: #USB,SET,OK

Команда \$KE,USB,GET

Синтаксис: \$KE,USB,GET

Чтение строки дескриптора USB устройства.

Ответ на запрос:

#USB,<Data> - чтение прошло успешно, поле *Data* содержит строковый дескриптор.

Пример:

Считать строковый дескриптор из памяти модуля:

запрос: \$KE,USB,GET
ответ: #USB, My USB Device

Команда \$KE,SER

Синтаксис: \$KE,SER

Чтение серийного номера модуля. Каждый модуль имеет свой собственный уникальный серийный номер.

Ответ на запрос:

#SER,<Serial Number>

Команда \$KE,RST

Синтаксис: \$KE,RST

Сброс всех настроек модуля в значение по умолчанию. При подаче команды все линии ввода/вывода устанавливаются на выход; на всех линиях устанавливается логический ноль; сохраненные настройки линий ввода/вывода в энергонезависимой памяти стираются и устанавливаются на выход; реле возвращаются в исходное состояние; стираются пользовательские данные; восстанавливается строковый дескриптор по умолчанию.

Ответ на запрос:

#RST,OK

KERNELCHIP

Компоненты для управления и мониторинга

Россия, Москва
+7 917 516-99-51

Mail: port@kernelchip.ru
Web: <http://www.kernelchip.ru>

