|  |  |  |
| --- | --- | --- |
| УДОСТОВЕРЕН |  |  |
| ЮФКВ.30139-01-УД |  |  |
|  |  |  |

Универсальные электронные модули УЭМ-МК, МВ98.03. Расширенная библиотека функций

Руководство программиста

ЮФКВ.30139-01 33 01

(ЮФКВ.30139-01 33 01-001ФЛ)

Листов 166

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

Литера

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Инв.№ подл. | Подп. и дата | | Взам.инв.№ | Инв.№ дубл. | Подп. и дата |
| 30875 | C:\Users\User\Desktop\podpic22.jpg | 19.08.2016 |  |  |  |

# АННОТАЦИЯ

Универсальные электронные модули мультиплексного канала совмещают функции контроллера шины, оконечного устройства, монитора шины, тестера протокола и имитатора дополнительных оконечных устройств на шине ГОСТ Р 52070-2003 [**1**].

Расширенная библиотека функций предназначена для разработки прикладных программ, использующих УЭМ-МК или МВ98.03 для решения задач тестирования и отладки интерфейсов ГОСТ Р 52070-2003 и систем на их основе, мониторинга и анализа процессов передачи данных по шине ГОСТ Р 52070-2003. Расширенная библиотека функций обеспечивает использование прикладными программами всех возможностей аппаратуры УЭМ-МК, МВ98.03, при этом управление аппаратурой производится в терминах протокола ГОСТ Р 52070-2003.

Настоящий документ является руководством программиста по расширенной библиотеке функций.

# СОДЕРЖАНИЕ

АННОТАЦИЯ 2

НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ 4

Указатель групп 5

Указатель структур данных 6

Список файлов 7

Группы 8

Вводные и дополнительные сведения 8

Коды завершения 8

Порядок действий при установлении связи с устройством 13

Адресная строка 14

Виртуальные устройства 14

Командные и ответные сегменты 14

Параметры передатчиков и характеристики выходных сигналов 15

Типы вносимых ошибок кодирования 17

Управление синхронизацией 21

Определения примитивных типов 23

Установление и разрыв связи с устройством 25

Действия с дескрипторами 28

Параметры конфигурации УЭМ 31

Описание параметров 32

Значения параметров 41

Параметры интервалов времени 55

Встроенный счетчик времени 58

Определения типов данных для КШ, ОУ, МШ 61

Функции КШ 68

Заполнение образа командного сегмента в ОЗУ ПЭВМ 68

Внесение ошибок состава сообщения 70

Создание и настройка командных сегментов 71

Создание и настройка кадров и программы КШ 81

Запуск и остановка КШ 87

Передача сообщений 88

Функции ОУ 90

Функции МШ 102

Запуск и остановка 104

Служебные функции 108

Формирование и разбор командных и ответных слов 111

Структуры данных 118

UEM\_BM\_MESSAGE 118

UEM\_CMD\_SEG 121

UEM\_DATA 123

UEM\_RAW\_BM\_MESSAGE 124

UEM\_RESP\_SEG 125

UEM\_SEGMENT\_DESCR 126

UEM\_TIME\_TAG 128

Файлы 130

uem.h 130

Алфавитный указатель 150

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ 164

ПЕРЕЧЕНЬ ССЫЛОЧНЫХ ДОКУМЕНТОВ 165

# НАЗНАЧЕНИЕ И УСЛОВИЯ ПРИМЕНЕНИЯ

Расширенная библиотека функций предназначена для разработки прикладных программ, использующих УЭМ-МК [**2**], МВ98.03 [**3**] для решения задач тестирования и отладки интерфейсов ГОСТ Р 52070-2003 [**1**] и систем на их основе, мониторинга и анализа процессов передачи данных по шине ГОСТ Р 52070-2003. Расширенная библиотека функций обеспечивает использование прикладными программами всех возможностей аппаратуры УЭМ-МК, МВ98.03, при этом управление аппаратурой производится в терминах протокола ГОСТ Р 52070-2003.

Разработка и эксплуатация прикладных программ, использующих расширенную библиотеку, должна выполняться на встроенной ПЭВМ в составе крейта VXI, в котором установлены модули УЭМ-МК или МВ98.03, не менее одного. На ПЭВМ должно быть установлено программное обеспечение VISA, unmbase [**4**] и unmuem [**5**, **6**].

Далее по тексту под аббревиатурой УЭМ мы будем понимать любой из модулей УЭМ-МК или МВ98.03.

# Указатель групп

## Группы

Полный список групп.

Вводные и дополнительные сведения

Коды завершения

Порядок действий при установлении связи с устройством

Адресная строка

Виртуальные устройства

Командные и ответные сегменты

Параметры передатчиков и характеристики выходных сигналов

Типы вносимых ошибок кодирования

Управление синхронизацией

Определения примитивных типов

Установление и разрыв связи с устройством

Действия с дескрипторами

Параметры конфигурации УЭМ

Описание параметров

Значения параметров

Параметры интервалов времени

Встроенный счетчик времени

Определения типов данных для КШ, ОУ, МШ

Функции КШ

Заполнение образа командного сегмента в ОЗУ ПЭВМ

Внесение ошибок состава сообщения

Создание и настройка командных сегментов

Создание и настройка кадров и программы КШ

Запуск и остановка КШ

Передача сообщений

Функции ОУ

Функции МШ

Запуск и остановка

Служебные функции

Формирование и разбор командных и ответных слов

# Указатель структур данных

## Структуры данных

Структуры данных с их кратким описанием.

**UEM\_BM\_MESSAGE (Разобранное сообщение МШ )**

**UEM\_CMD\_SEG (Образ командного сегмента )**

**UEM\_DATA (Блок слов данных )**

**UEM\_RAW\_BM\_MESSAGE (Принятое сообщение в аппаратном формате )**

**UEM\_RESP\_SEG (Ответный сегмент )**

**UEM\_SEGMENT\_DESCR (Описатель сегмента в мониторе шины )**

**UEM\_TIME\_TAG (Формат метки времени )**

# Список файлов

## Файлы

Полный список файлов.

**uem.h (Универсальные электронные модули УЭМ-МК, МВ98.03. Расширенная библиотека функций. Файл заголовков функций )**

# Группы

## Вводные и дополнительные сведения

Введение терминов, необходимые сведения об основных возможностях аппаратуры, типовые сценарии.

### Группы

* **Коды завершения**

*Результаты выполнения функций.*

* **Порядок действий при установлении связи с устройством**

*Описание типового сценария*

* **Адресная строка**

*Синтаксис идентификатора инструмента в VISA.*

* **Виртуальные устройства**

*Введение термина: виртуальное устройство*

* **Командные и ответные сегменты**

*Введение терминов: сегмент, командный сегмент, ответный сегмент*

* **Параметры передатчиков и характеристики выходных сигналов**

*Сведения о возможностях аппаратного обеспечения. Соответствие значений параметров конфигурации передатчиков и электрических характеристик выходных сигналов.*

* **Типы вносимых ошибок кодирования**

*Сведения о возможностях аппаратного обеспечения в части внесения ошибок кодирования и способы описания (программирования) вносимых ошибок.*

* **Управление синхронизацией**

*Сведения об аппаратных средствах синхронизации с внешним оборудованием и об управлении ими.*

### Подробное описание

Введение терминов, необходимые сведения об основных возможностях аппаратуры, типовые сценарии.

## Коды завершения

Результаты выполнения функций.

### Макросы

* #define **UEM\_WARN\_OFFSET**  (0x3FFC0B00L)

*Начальный номер кодов предупреждений.*

* #define **UEM\_ERROR\_OFFSET**  (\_VI\_ERROR + **UEM\_WARN\_OFFSET**)

*Начальный номер кодов ошибок.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE**  (**UEM\_ERROR\_OFFSET** + 0)

*Недопустимое значение параметра.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_1**  (**UEM\_ERROR\_OFFSET** + 1)

*Недопустимое значение в параметре 1.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_2**  (**UEM\_ERROR\_OFFSET** + 2)

*Недопустимое значение в параметре 2.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_3**  (**UEM\_ERROR\_OFFSET** + 3)

*Недопустимое значение в параметре 3.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_4**  (**UEM\_ERROR\_OFFSET** + 4)

*Недопустимое значение в параметре 4.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_5**  (**UEM\_ERROR\_OFFSET** + 5)

*Недопустимое значение в параметре 5.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_6**  (**UEM\_ERROR\_OFFSET** + 6)

*Недопустимое значение в параметре 6.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_7**  (**UEM\_ERROR\_OFFSET** + 7)

*Недопустимое значение в параметре 7.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_8**  (**UEM\_ERROR\_OFFSET** + 8)

*Недопустимое значение в параметре 8.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_9**  (**UEM\_ERROR\_OFFSET** + 9)

*Недопустимое значение в параметре 9.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_10**  (**UEM\_ERROR\_OFFSET** + 10)

*Недопустимое значение в параметре 10.*

* #define **UEM\_ERROR\_INV\_HANDLE**  (**UEM\_ERROR\_OFFSET** + 11)

*Недействительный дескриптор.*

* #define **UEM\_ERROR\_INV\_HANDLE\_TYPE**  (**UEM\_ERROR\_OFFSET** + 12)

*Неподходящий тип дескриптора.*

* #define **UEM\_ERROR\_NO\_FREE\_RAM**  (**UEM\_ERROR\_OFFSET** + 13)

*Недостаточно ОЗУ УЭМ.*

* #define **UEM\_ERROR\_NO\_HOST\_MEM**  (**UEM\_ERROR\_OFFSET** + 14)

*Недостаточно ОЗУ управляющей ПЭВМ.*

* #define **UEM\_ERROR\_NOT\_CONNECTED**  (**UEM\_ERROR\_OFFSET** + 15)

*Нет связи с устройством.*

* #define **UEM\_ERROR\_INPOOL**  (**UEM\_ERROR\_OFFSET** + 16)

*Внутренняя ошибка менеджера памяти.*

* #define **UEM\_ERROR\_BCP\_NINST**  (**UEM\_ERROR\_OFFSET** + 17)

*Не установлена программа КШ.*

* #define **UEM\_ERROR\_FORMAT\_DISABLED**  (**UEM\_ERROR\_OFFSET** + 18)

*Формат сообщения запрещен конфигурацией УЭМ.*

* #define **UEM\_ERROR\_FORMAT\_X\_MCODE**  (**UEM\_ERROR\_OFFSET** + 19)

*Формат сообщения несовместим с командой управления.*

* #define **UEM\_ERROR\_NOT\_APPLICABLE**  (**UEM\_ERROR\_OFFSET** + 20)

*Действие не применимо к объекту.*

* #define **UEM\_ERROR\_ADDRESS\_OUT\_OF\_RANGE**  (**UEM\_ERROR\_OFFSET** + 21)

*Адрес вне допустимого диапазона.*

* #define **UEM\_ERROR\_NUMBER\_OUT\_OF\_RANGE**  (**UEM\_ERROR\_OFFSET** + 23)

*Номер вне допустимого диапазона.*

* #define **UEM\_ERROR\_BAD\_TIMEOUT**  (**UEM\_ERROR\_OFFSET** + 24)

*Недопустимое значение таймаута*

* #define **UEM\_ERROR\_BAD\_OVERLAY\_SOURCE**  (**UEM\_ERROR\_OFFSET** + 25)

*Недопустимые исходные сегменты для наложения.*

* #define **UEM\_ERROR\_WRONG\_LOCATION**  (**UEM\_ERROR\_OFFSET** + 26)

*Объект расположен не в том устройстве.*

* #define **UEM\_ERROR\_TOO\_MANY\_DATAWORDS**  (**UEM\_ERROR\_OFFSET** + 27)

*Слишком много слов данных.*

* #define **UEM\_ERROR\_MAX\_SIZE\_EXCEED**  (**UEM\_ERROR\_OFFSET** + 28)

*Превышен максимальный размер.*

* #define **UEM\_ERROR\_NO\_FRAME\_APPEND**  (**UEM\_ERROR\_OFFSET** + 29)

*Не добавлен кадр.*

* #define **UEM\_ERROR\_IN\_USE**  (**UEM\_ERROR\_OFFSET** + 30)

*Устройство или объект используются.*

* #define **UEM\_ERROR\_THREAD\_FAULT**  (**UEM\_ERROR\_OFFSET** + 31)

*Ошибки в работе служебной нити.*

* #define **UEM\_ERROR\_BM\_INTERNAL\_BUFFER\_OVERFLOW**  (**UEM\_ERROR\_OFFSET** + 32)

*Переполнение внутреннего буфера МШ.*

* #define **UEM\_ERROR\_INC\_RESP**  (**UEM\_ERROR\_OFFSET** + 33)

*Несовместимый ответный сегмент.*

* #define **UEM\_WARN\_NO\_NEXT\_MESSAGE**  (**UEM\_WARN\_OFFSET** + 0)

*Нет следующего сообщения (в буфере МШ).*

* #define **UEM\_WARN\_JUST\_IN\_STATE**  (**UEM\_WARN\_OFFSET** + 1)

*Устройство уже в нужном состоянии; никаких действий не производится.*

### Подробное описание

Результаты выполнения функций.

Нулевой код завершения соответствует нормальному завершению. Коды, меньшие 0, являются кодами ошибок. Коды, большие 0, являются кодами предупреждений и особенных ситуаций, не являющихся ошибками.

Следует иметь в виду, что код завершения может быть сформирован не только настоящей расширенной библиотекой, но и нижележащем ПО: драйвером УЭМ, драйвером носителя мезонинов, ПО VISA. По поводу описания смысла этих кодов завершения отсылаем к документации по указанному ПО [**4**,**5**,**6**].

### Макросы

#### #define UEM\_WARN\_OFFSET  (0x3FFC0B00L)

Начальный номер кодов предупреждений.

#### #define UEM\_ERROR\_OFFSET  (\_VI\_ERROR + UEM\_WARN\_OFFSET)

Начальный номер кодов ошибок.

#### #define UEM\_ERROR\_BAD\_PARAM\_VALUE  (UEM\_ERROR\_OFFSET + 0)

Недопустимое значение параметра.

#### #define UEM\_ERROR\_BAD\_PARAM\_VALUE\_1  (UEM\_ERROR\_OFFSET + 1)

Недопустимое значение в параметре 1.

#### #define UEM\_ERROR\_BAD\_PARAM\_VALUE\_2  (UEM\_ERROR\_OFFSET + 2)

Недопустимое значение в параметре 2.

#### #define UEM\_ERROR\_BAD\_PARAM\_VALUE\_3  (UEM\_ERROR\_OFFSET + 3)

Недопустимое значение в параметре 3.

#### #define UEM\_ERROR\_BAD\_PARAM\_VALUE\_4  (UEM\_ERROR\_OFFSET + 4)

Недопустимое значение в параметре 4.

#### #define UEM\_ERROR\_BAD\_PARAM\_VALUE\_5  (UEM\_ERROR\_OFFSET + 5)

Недопустимое значение в параметре 5.

#### #define UEM\_ERROR\_BAD\_PARAM\_VALUE\_6  (UEM\_ERROR\_OFFSET + 6)

Недопустимое значение в параметре 6.

#### #define UEM\_ERROR\_BAD\_PARAM\_VALUE\_7  (UEM\_ERROR\_OFFSET + 7)

Недопустимое значение в параметре 7.

#### #define UEM\_ERROR\_BAD\_PARAM\_VALUE\_8  (UEM\_ERROR\_OFFSET + 8)

Недопустимое значение в параметре 8.

#### #define UEM\_ERROR\_BAD\_PARAM\_VALUE\_9  (UEM\_ERROR\_OFFSET + 9)

Недопустимое значение в параметре 9.

#### #define UEM\_ERROR\_BAD\_PARAM\_VALUE\_10  (UEM\_ERROR\_OFFSET + 10)

Недопустимое значение в параметре 10.

#### #define UEM\_ERROR\_INV\_HANDLE  (UEM\_ERROR\_OFFSET + 11)

Недействительный дескриптор.

#### #define UEM\_ERROR\_INV\_HANDLE\_TYPE  (UEM\_ERROR\_OFFSET + 12)

Неподходящий тип дескриптора.

#### #define UEM\_ERROR\_NO\_FREE\_RAM  (UEM\_ERROR\_OFFSET + 13)

Недостаточно ОЗУ УЭМ.

#### #define UEM\_ERROR\_NO\_HOST\_MEM  (UEM\_ERROR\_OFFSET + 14)

Недостаточно ОЗУ управляющей ПЭВМ.

#### #define UEM\_ERROR\_NOT\_CONNECTED  (UEM\_ERROR\_OFFSET + 15)

Нет связи с устройством.

#### #define UEM\_ERROR\_INPOOL  (UEM\_ERROR\_OFFSET + 16)

Внутренняя ошибка менеджера памяти.

#### #define UEM\_ERROR\_BCP\_NINST  (UEM\_ERROR\_OFFSET + 17)

Не установлена программа КШ.

#### #define UEM\_ERROR\_FORMAT\_DISABLED  (UEM\_ERROR\_OFFSET + 18)

Формат сообщения запрещен конфигурацией УЭМ.

#### #define UEM\_ERROR\_FORMAT\_X\_MCODE  (UEM\_ERROR\_OFFSET + 19)

Формат сообщения несовместим с командой управления.

#### #define UEM\_ERROR\_NOT\_APPLICABLE  (UEM\_ERROR\_OFFSET + 20)

Действие не применимо к объекту.

В связи с особенностями данного объекта.

#### #define UEM\_ERROR\_ADDRESS\_OUT\_OF\_RANGE  (UEM\_ERROR\_OFFSET + 21)

Адрес вне допустимого диапазона.

#### #define UEM\_ERROR\_NUMBER\_OUT\_OF\_RANGE  (UEM\_ERROR\_OFFSET + 23)

Номер вне допустимого диапазона.

#### #define UEM\_ERROR\_BAD\_TIMEOUT  (UEM\_ERROR\_OFFSET + 24)

Недопустимое значение таймаута.

#### #define UEM\_ERROR\_BAD\_OVERLAY\_SOURCE  (UEM\_ERROR\_OFFSET + 25)

Недопустимые исходные сегменты для наложения.

Сегменты не являются обычными командными сегментами или привязаны к одной и той же шине.

#### #define UEM\_ERROR\_WRONG\_LOCATION  (UEM\_ERROR\_OFFSET + 26)

Объект расположен не в том устройстве.

#### #define UEM\_ERROR\_TOO\_MANY\_DATAWORDS  (UEM\_ERROR\_OFFSET + 27)

Слишком много слов данных.

#### #define UEM\_ERROR\_MAX\_SIZE\_EXCEED  (UEM\_ERROR\_OFFSET + 28)

Превышен максимальный размер.

#### #define UEM\_ERROR\_NO\_FRAME\_APPEND  (UEM\_ERROR\_OFFSET + 29)

Не добавлен кадр.

#### #define UEM\_ERROR\_IN\_USE  (UEM\_ERROR\_OFFSET + 30)

Устройство или объект используются.

#### #define UEM\_ERROR\_THREAD\_FAULT  (UEM\_ERROR\_OFFSET + 31)

Ошибки в работе служебной нити.

#### #define UEM\_ERROR\_BM\_INTERNAL\_BUFFER\_OVERFLOW  (UEM\_ERROR\_OFFSET + 32)

Переполнение внутреннего буфера МШ.

#### #define UEM\_ERROR\_INC\_RESP  (UEM\_ERROR\_OFFSET + 33)

Несовместимый ответный сегмент.

#### #define UEM\_WARN\_NO\_NEXT\_MESSAGE  (UEM\_WARN\_OFFSET + 0)

Нет следующего сообщения (в буфере МШ).

#### #define UEM\_WARN\_JUST\_IN\_STATE  (UEM\_WARN\_OFFSET + 1)

Устройство уже в нужном состоянии; никаких действий не производится.

## Порядок действий при установлении связи с устройством

Описание типового сценария.

Для установления связи с устройством необходимо:

1. Открыть сеанс связи с носителем мезонинов при помощи функции **unmbase\_init()** [**4**].

2. Открыть сеанс связи с устройством УЭМ при помощи функции **uem\_init()**.

3. Связать эти сеансы при помощи функции **uem\_connect()**.

Например:

**unmbase\_init** ("VXI0::1::INSTR", VI\_TRUE, VI\_TRUE, &vi);

**uem\_init** ("VXI0::1::INSTR", VI\_TRUE, VI\_TRUE, &uem);

**uem\_connect** (uem, vi, num, VI\_TRUE, VI\_TRUE);

## Адресная строка

Синтаксис идентификатора инструмента в VISA.

Адресная строка идентифицирует инструмент в системе VISA.

Для шины VXI строка имеет вид:

**VXI** [board]::VXI-la[::INSTR]

Где **board** - номер интерфейса VXI (по умолчанию 0), **VXI-la** - логический адрес VXI.

Необязательные параметры указаны в квадратных скобках.

## Виртуальные устройства

Введение термина: виртуальное устройство.

В составе УЭМ имеется аппаратура, выполняющая функции контроллера шины (КШ), оконечных устройств (ОУ), монитора шины (МШ). При конфигурировании и управлении УЭМ удобно рассматривать эти компоненты аппаратуры УЭМ как отдельные устройства. В настоящей библиотеке и настоящем руководстве такие устройства называются *виртуальными устройствами* в составе УЭМ.

Для управления виртуальными устройствами их сначала надо открыть. Это выполняется при помощи функций **uem\_bc\_init()**, **uem\_rt\_init()**, **uem\_bm\_init()**. Открывать виртуальные устройства возможно только после того, как будет открыт УЭМ (см. **Порядок действий при установлении связи с устройством**).

Допускается открыть один виртуальный КШ, один виртуальный МШ, до 31 виртуальных ОУ (до 32 виртуальных ОУ при запрете групповых сообщений).

## Командные и ответные сегменты

Введение терминов: сегмент, командный сегмент, ответный сегмент.

Под ***сегментом*** понимается последовательность соприкасающихся слов, то есть слов, передаваемых по МКПД без пауз.

Сегмент, передаваемый контроллером шины, содержит в начале командное слово (в форматах 3, 8 - два командных слова), за которым, в зависимости от формата сообщения МКПД, могут следовать слова данных. Сегмент, передаваемый контроллером шины, в настоящей библиотеке и настоящем документе называется ***командным сегментом***.

Сегмент, передаваемый оконечным устройством, состоит из ответного слова, за которым, в зависимости от формата сообщения МКПД, могут следовать слова данных. Сегмент, передаваемый оконечным устройством, в настоящей библиотеке и настоящем документе называется ***ответным сегментом***.

В терминах сегментов структура сообщений МКПД описывается следующим образом:

Сообщение состоит из командного сегмента, передаваемого контроллером шины, и ответных сегментов, передаваемых в ответ оконечными устройствами. В зависимости от формата сообщение может включать ноль, один или два ответных сегмента.

Более подробную информацию о форматах сообщений МКПД можно получить в [**1**].

## Параметры передатчиков и характеристики выходных сигналов

Сведения о возможностях аппаратного обеспечения. Соответствие значений параметров конфигурации передатчиков и электрических характеристик выходных сигналов.

1. **Приблизительное соответствие значений параметров UEM\_TXA\_RFT и UEM\_TXB\_RFT и длительностей фронта и среза сигнала МКПД.**

2. **Приблизительное соответствие значений параметров UEM\_TXA\_VPP и UEM\_TXB\_VPP и размаха сигнала МКПД.**

Таблица 1. **Приблизительное соответствие значений параметров UEM\_TXA\_RFT и UEM\_TXB\_RFT и длительностей фронта и среза сигнала МКПД**

Разряды [15:6] доступны по записи и по чтению. Значение по умолчанию 0018h. Вводимое значение параметра имеет следующие особенности:

- два младших значения (0000h, 0001h) интерпретируются аппаратурой как 0002h;

- все нечетные значения из диапазона от 0003h до 0031h включительно интерпретируются аппаратурой как предшествующие четные значения (0003h как 0002h, 0005h как 0004h, и т.д.);

- максимальное значение составляет 0032h, все значения от 0033h и более интерпретируются аппаратурой как 0032h.

| **Значение параметра (hex)** | **Характеристика сигнала**  Длительность фронта (среза) импульсов сигнала на выходе устройства, нс (погрешность фактического значения не превышает 5 %) |
| --- | --- |
| 0002 | 60 |
| 0004 | 65 |
| 0006 | 75 |
| 0008 | 85 |
| 000A | 95 |
| 000C | 110 |
| 000E | 120 |
| 0010 | 135 |
| 0012 | 150 |
| 0014 | 165 |
| 0016 | 180 |
| 0018 | 195 |
| 001A | 210 |
| 001C | 225 |
| 001E | 240 |
| 0020 | 255 |
| 0022 | 270 |
| 0024 | 285 |
| 285 | 300 |
| 0028 | 315 |
| 002A | 330 |
| 002C | 350 |
| 002E | 365 |
| 0030 | 370 |
| 0032 | Отдельная настройка передатчика, при которой форма сигнала при длительности импульсов 500 нс аппроксимирует синусоидальный сигнал |

Таблица 2. **Приблизительное соответствие значений параметров UEM\_TXA\_VPP и UEM\_TXB\_VPP и размаха сигнала МКПД**

| **Значение параметра (hex)** | **Характеристика сигнала**  (НП – непосредственное подключение, ТП – трансформаторное подключение по ГОСТ Р 52070)  Размах сигнала на выходе устройства на эквивалентной нагрузке 35 Ом (НП) / 70 Ом (ТП), В: |
| --- | --- |
| 0000 … 000D | 0 … 0,28 (НП) / 0 … 0,8 (ТП), при установке таких значений следует учитывать, что сигнал может находиться ниже границы порога срабатывания приемника, и информация может не быть принята МШ |
| 000E / 000E | 0,28 (НП) / 0,8 (ТП, соответствует 0,2 В в точке подключения ТМ по ГОСТ Р 51765) |
| 003B / 003A | 1,2 (НП) / 3,44 (ТП, соответствует 0,86 В в точке подключения ТМ по ГОСТ Р 51765) |
| 0090 / 0090 | 3,0 (НП) / 8,4 (ТП, соответствует 2,1 В в точке подключения ТМ по ГОСТ Р 51765) |
| 0136 / 0136 | 6,38 (НП) / 18,3 (ТП), значение по умолчанию |
| 01B6 / 0195 | 9,0 (НП) / 24,0 (ТП, соответствует 6,0 В в точке подключения ТМ по ГОСТ Р 51765) |
| 01FF | 10,8 (НП) / 30,8 (ТП), максимальное значение |

## Типы вносимых ошибок кодирования

Сведения о возможностях аппаратного обеспечения в части внесения ошибок кодирования и способы описания (программирования) вносимых ошибок.

### Перечисления

* enum **UEM\_ERROR\_TYPE** { **UEM\_ERRT\_NONE** = 0, **UEM\_ERRT\_INV\_PARITY** = 1, **UEM\_ERRT\_WRONG\_BITCOUNT** = 2, **UEM\_ERRT\_BAD\_SYNCHRO** = 3, **UEM\_ERRT\_BAD\_BIPHASE\_ZERO** = 4, **UEM\_ERRT\_BAD\_BIPHASE\_POS** = 5, **UEM\_ERRT\_BAD\_BIPHASE\_NEG** = 6, **UEM\_ERRT\_SHIFT\_EDGE** = 7 }

*Тип вносимой ошибки кодирования.*

* enum **UEM\_SYNCHRO\_ERROR\_POS** { **UEM\_BAD\_SYNCHRO\_NONE** = 0, **UEM\_BAD\_SYNCHRO\_IEEEEE** = 1, **UEM\_BAD\_SYNCHRO\_EIEEEE** = 2, **UEM\_BAD\_SYNCHRO\_EEIEEE** = 3, **UEM\_BAD\_SYNCHRO\_NONE2** = 4, **UEM\_BAD\_SYNCHRO\_EEEIEE** = 5, **UEM\_BAD\_SYNCHRO\_EEEEIE** = 6, **UEM\_BAD\_SYNCHRO\_EEEEEI** = 7 }

*Позиция ошибки кодирования синхроимпульса.*

### Граничные значения изменения количества разрядов

Эти константы задают граничные значения аргумента **error\_pos** в функциях **uem\_cseg\_error\_set()**, **uem\_response\_error\_set()**, когда в аргументе **error\_type** указано **UEM\_ERRT\_WRONG\_BITCOUNT**.

* #define **UEM\_BITCOUNT\_CHANGE\_MIN**  (-3)

*Минимальное приращение количества разрядов.*

* #define **UEM\_BITCOUNT\_CHANGE\_MAX**  (+3)

*Максимальное приращение количества разрядов.*

### Граничные значения позиции ошибки

Эти константы задают граничные значения аргумента **error\_pos** для некоторых значений аргумента **error\_type** в функциях **uem\_cseg\_error\_set()**, **uem\_response\_error\_set()**.

* #define **UEM\_BIPHASE\_POS\_MIN**  4

*Минимальная позиция при* ***error\_type****, равном* ***UEM\_ERRT\_BAD\_BIPHASE\_ZERO****,* ***UEM\_ERRT\_BAD\_BIPHASE\_POS*** *или* ***UEM\_ERRT\_BAD\_BIPHASE\_NEG****.*

* #define **UEM\_BIPHASE\_POS\_MAX**  20

*Максимальная позиция при* ***error\_type****, равном* ***UEM\_ERRT\_BAD\_BIPHASE\_ZERO****,* ***UEM\_ERRT\_BAD\_BIPHASE\_POS*** *или* ***UEM\_ERRT\_BAD\_BIPHASE\_NEG****.*

* #define **UEM\_SHIFT\_POS\_MIN**  0

*Минимальная позиция при* ***error\_type****, равном* ***UEM\_ERRT\_SHIFT\_EDGE****.*

* #define **UEM\_SHIFT\_POS\_MAX**  40

*Максимальная позиция при* ***error\_type****, равном* ***UEM\_ERRT\_SHIFT\_EDGE****.*

### Граничные значения величины сдвига

Эти константы задают граничные значения для аргумента **error\_param** в функциях **uem\_cseg\_error\_set()**, **uem\_response\_error\_set()**, когда аргумент **error\_type** равен **UEM\_ERRT\_SHIFT\_EDGE**. Величина сдвига указывается в единицах по 10 нс. Отрицательные значения обозначают сдвиг влево, положительные - вправо.

* #define **UEM\_SHIFT\_LENGTH\_MIN**  (-25)

*Максимальная величина сдвига влево.*

* #define **UEM\_SHIFT\_LENGTH\_MAX**  (+25)

*Максимальная величина сдвига вправо.*

### Значения аргументов по умолчанию

Эти константы задают значения аргументов функций **uem\_cseg\_error\_set()**, **uem\_response\_error\_set()** по умолчанию, соответствуют отсутствию внесения ошибок.

* #define **UEM\_ERROR\_TYPE\_DEFAULT**  0

*Значение по умолчанию для* ***error\_type****.*

* #define **UEM\_ERROR\_POS\_DEFAULT**  0

*Значение по умолчанию для* ***error\_pos****.*

* #define **UEM\_ERROR\_PARAM\_DEFAULT**  0

*Значение по умолчанию для* ***error\_param****.*

### Подробное описание

Сведения о возможностях аппаратного обеспечения в части внесения ошибок кодирования и способы описания (программирования) вносимых ошибок.

Программирование ошибок кодирования выполняется при помощи функций **uem\_cseg\_error\_set()**, **uem\_response\_error\_set()**, имеющих для описания ошибок следующие параметры: **error\_type** - основной параметр - тип вносимой ошибки, **error\_pos** и **error\_param** - дополнительные параметры, значения которых интерпретируются в зависимости от типа ошибки. Параметр **error\_type** должен принимать значения из перечисления **UEM\_ERROR\_TYPE**. Описания типов ошибок и интерпретации дополнительных параметров содержится в описании этого перечисления. См. **UEM\_ERROR\_TYPE**.

В данном разделе также определены константы для задания параметров **error\_pos** и **error\_param** при различных **error\_type**.

### Макросы

#### #define UEM\_BITCOUNT\_CHANGE\_MIN  (-3)

Минимальное приращение количества разрядов.

#### #define UEM\_BITCOUNT\_CHANGE\_MAX  (+3)

Максимальное приращение количества разрядов.

#### #define UEM\_BIPHASE\_POS\_MIN  4

Минимальная позиция при **error\_type**, равном **UEM\_ERRT\_BAD\_BIPHASE\_ZERO**, **UEM\_ERRT\_BAD\_BIPHASE\_POS** или **UEM\_ERRT\_BAD\_BIPHASE\_NEG**.

#### #define UEM\_BIPHASE\_POS\_MAX  20

Максимальная позиция при **error\_type**, равном **UEM\_ERRT\_BAD\_BIPHASE\_ZERO**, **UEM\_ERRT\_BAD\_BIPHASE\_POS** или **UEM\_ERRT\_BAD\_BIPHASE\_NEG**.

#### #define UEM\_SHIFT\_POS\_MIN  0

Минимальная позиция при **error\_type**, равном **UEM\_ERRT\_SHIFT\_EDGE**.

#### #define UEM\_SHIFT\_POS\_MAX  40

Максимальная позиция при **error\_type**, равном **UEM\_ERRT\_SHIFT\_EDGE**.

#### #define UEM\_SHIFT\_LENGTH\_MIN  (-25)

Максимальная величина сдвига влево.

#### #define UEM\_SHIFT\_LENGTH\_MAX  (+25)

Максимальная величина сдвига вправо.

#### #define UEM\_ERROR\_TYPE\_DEFAULT  0

Значение по умолчанию для **error\_type**.

#### #define UEM\_ERROR\_POS\_DEFAULT  0

Значение по умолчанию для **error\_pos**.

#### #define UEM\_ERROR\_PARAM\_DEFAULT  0

Значение по умолчанию для **error\_param**.

### Перечисления

#### enum UEM\_ERROR\_TYPE

Тип вносимой ошибки кодирования.

**Элементы перечислений:**

***UEM\_ERRT\_NONE*** Ошибка не вносится.

***UEM\_ERRT\_INV\_PARITY*** Инверсия разряда четности (20-го) по отношению к его достоверному значению.

***UEM\_ERRT\_WRONG\_BITCOUNT*** Ошибка количества разрядов в слове.

Параметр **error\_pos** интерпретируется как изменение числа разрядов. Допустимые значения: от -3 (**UEM\_BITCOUNT\_CHANGE\_MIN**) до +3 (**UEM\_BITCOUNT\_CHANGE\_MAX**).

При уменьшении количества разрядов последовательно исключаются 19-й, 18-й, 17-й разряды 20-ти разрядного слова по ГОСТ Р 52070, при увеличении – «избыточные» разряды добавляются после 19-го разряда, «сдвигая» разряд четности на последнюю позицию, при этом бит четности всегда подсчитывается с учетом заданных значений «избыточных» разрядов, либо с учетом только «оставшихся» разрядов при «укорочении», по правилу дополнения до нечетности.

Параметр **error\_param** при **error\_pos** > 0 содержит значения для дополнительных разрядов слова, бит 0 задает 20-й разряд слова, бит 1 - 21-й (если он есть), бит 2 - 22-й (если он есть).

***UEM\_ERRT\_BAD\_SYNCHRO*** Ошибка кодирования синхроимпульса слова.

Параметр **error\_pos** должен содержать код позиции ошибки в синхроимпульсе, один из элементов перечисления **UEM\_SYNCHRO\_ERROR\_POS**.

***UEM\_ERRT\_BAD\_BIPHASE\_ZERO*** Пропуск перехода через 0, удерживается нулевой уровень.

Ошибка бифазного кодирования типа «проскок» в выбранном разряде слова, т.е. установка выходного сигнала МКПД в нулевой уровень напряжения в течение интервала передачи выбранного разряда.

Ошибка вносится в разряд с номером, указанном в параметре **error\_pos**. Допустимый диапазон значений: от 4 (**UEM\_BIPHASE\_POS\_MIN**) до 20 (**UEM\_BIPHASE\_POS\_MAX**).

***UEM\_ERRT\_BAD\_BIPHASE\_POS*** Пропуск перехода через 0, удерживается положительный уровень.

Ошибка бифазного кодирования, т.е. отсутствие обязательного перехода через 0 выходного сигнала МКПД в середине выбранного разряда, с установкой положительного уровня напряжения в течение интервала передачи выбранного разряда.

Ошибка вносится в разряд с номером, указанном в параметре **error\_pos**. Допустимый диапазон значений: от 4 (**UEM\_BIPHASE\_POS\_MIN**) до 20 (**UEM\_BIPHASE\_POS\_MAX**).

***UEM\_ERRT\_BAD\_BIPHASE\_NEG*** Пропуск перехода через 0, удерживается отрицательный уровень.

Ошибка бифазного кодирования, т.е. отсутствие обязательного перехода через 0 выходного сигнала МКПД в середине выбранного разряда, с установкой отрицательного уровня напряжения в течение интервала передачи выбранного разряда.

Ошибка вносится в разряд с номером, указанном в параметре **error\_pos**. Допустимый диапазон значений: от 4 (**UEM\_BIPHASE\_POS\_MIN**) до 20 (**UEM\_BIPHASE\_POS\_MAX**).

***UEM\_ERRT\_SHIFT\_EDGE*** Сдвиг перехода через 0.

Ошибка смещения момента пересечения нулевого уровня напряжения для выбранного перехода через 0 на заданную величину.

Параметр **error\_pos** задает позицию сдвигаемого перехода - номер полуразряда слова, допустимые значения от 0 (**UEM\_SHIFT\_POS\_MIN**) до 40 (**UEM\_SHIFT\_POS\_MAX**).

Параметр **error\_param** задает направление («влево» или «вправо») и величину временного сдвига с разрешением 10 нс на МЗР в диапазоне от -25 (**UEM\_SHIFT\_LENGTH\_MIN**) до +25 (**UEM\_SHIFT\_LENGTH\_MAX**), т.е. от -250 до +250 нс.

Технический прием внесения данной ошибки заключается в инверсии (одновременной смене полярности сигналов управления передатчиком МКПД относительно «номинальных» логических уровней) выходного сигнала на следующих интервалах:

– при сдвиге «влево»: от точки (((**error\_pos** – 1) x 500) + (500 – (**error\_param** x 10))) нс до точки (**error\_pos** x 500) нс;

– при сдвиге «вправо»: от точки (**error\_pos** x 500) нс до точки ((**error\_pos** x 500) + (**error\_param** x 10)) нс.

Инверсия производится по описанным правилам вне зависимости от того, присутствует ли в точке, заданной значением параметра **error\_pos**, пересечение нуля; если в «номинальном» коде пересечения нуля в данной точке нет, то при внесении ошибки будет сформирован «лишний» переход через ноль.

#### enum UEM\_SYNCHRO\_ERROR\_POS

Позиция ошибки кодирования синхроимпульса.

Эти значения предназначены для использования в аргументе **error\_pos** в функциях **uem\_cseg\_error\_set()**, **uem\_response\_error\_set()**, когда в аргументе **error\_type** указано **UEM\_ERRT\_BAD\_SYNCHRO**.

Под 1/6 частями синхроимпульса подразумеваются 6 последовательных интервалов по 500 нс, образующих этот синхроимпульс. Инверсия подразумевает одновременную смену полярности сигналов управления передатчиком МКПД на заданном интервале относительно «номинальных» уровней данных сигналов на этом интервале.

**Элементы перечислений:**

***UEM\_BAD\_SYNCHRO\_NONE*** Ошибка не вносится.

***UEM\_BAD\_SYNCHRO\_IEEEEE*** Инверсия первой 1/6.

***UEM\_BAD\_SYNCHRO\_EIEEEE*** Инверсия второй 1/6.

***UEM\_BAD\_SYNCHRO\_EEIEEE*** Инверсия третьей 1/6.

***UEM\_BAD\_SYNCHRO\_NONE2*** Ошибка не вносится.

***UEM\_BAD\_SYNCHRO\_EEEIEE*** Инверсия четвертой 1/6.

***UEM\_BAD\_SYNCHRO\_EEEEIE*** Инверсия пятой 1/6.

***UEM\_BAD\_SYNCHRO\_EEEEEI*** Инверсия шестой 1/6.

## Управление синхронизацией

Сведения об аппаратных средствах синхронизации с внешним оборудованием и об управлении ими.

В УЭМ существует четыре сигнала внешней синхронизации [**2**, **3**]:

##### sync\_in\_1

Входной сигнал внешней синхронизации 1.

##### sync\_in\_2

Входной сигнал внешней синхронизации 2.

##### sync\_out\_1

Выходной сигнал внешней синхронизации 1.

##### sync\_out\_2

Выходной сигнал внешней синхронизации 2.

Сигналы **sync\_in\_1** и **sync\_in\_2** используются для синхронизации КШ с внешним оборудованием.

Сигнал **sync\_in\_1** служит для выдачи контроллером шины сообщений или слов в линию синхронно с внешним источником синхронизации. Такой синхронизации подвергаются сообщения или слова, для которых при программировании паузы функциями **uem\_cseg\_gap\_get()**, **uem\_cseg\_word\_gap\_set()** в аргументе **flags** указан флаг **UEM\_CSEG\_GAP\_ESYNC**. При функционировании КШ сигнал может использоваться как для задания периода всех сообщений (если привязка установлена в каждом КС), так и для задания периода блоков сообщений (кадров) (если привязка установлена только в первом КС блока).

Сигнал **sync\_in\_2** служит для отложенного или периодического запуска КШ, при запуске КШ функцией **uem\_bc\_start()** с аргументом **flags** = **UEM\_BC\_START\_WAITING**.

Сигналы **sync\_out\_1** и **sync\_out\_2** формируются МШ и сообщают внешнему оборудованию о событиях в шине, детектируемых при помощи ряда условий.

Сигнал **sync\_out\_1** формируется после того, как обнаружена передача по шине МКПД слова, для которого выполняются каждое из следующих условий:

* Слово передано по шине, определяемой параметром конфигурации **UEM\_SYNC1\_CH**.
* Перед словом была пауза в передаче или не было такой паузы - в соответствии с заданным значением параметра **UEM\_SYNC1\_GAPB**.
* Передача слова выполнена с ошибкой или без ошибки в соответствии с заданным значением параметра **UEM\_SYNC1\_ERR**.
* Синхроимпульс слова соответствует заданному значению параметра конфигурации **UEM\_SYNC1\_C\_D\_**.
* Содержание (информационных разрядов) слова идентично значению параметра **UEM\_BM\_WORD\_PATTERN** (при этом в сравнении участвуют только разряды, заданные параметром **UEM\_BM\_WORD\_MASK**).

Сигнал **sync\_out\_2** формируется после обнаружения в шине достоверного командного или ответного слова с адресом ОУ, определяемым параметром конфигурации **UEM\_SYNC2\_VRTA**.

Обработка и формирование сигналов синхронизации разрешается параметрами конфигурации **UEM\_SYNC\_IN\_1\_ENA**, **UEM\_SYNC\_IN\_2\_ENA**, **UEM\_SYNC\_OUT\_1\_ENA**, **UEM\_SYNC\_OUT\_2\_ENA** соответственно. По умолчанию обработка и формирование запрещены.

Однократная программная имитация поступления или формирования сигналов выполняется путем записи значения 1 в параметры конфигурации **UEM\_SYNC\_IN\_1\_SET**, **UEM\_SYNC\_IN\_2\_SET**, **UEM\_SYNC\_OUT\_1\_SET**, **UEM\_SYNC\_OUT\_2\_SET** соответственно.

Для сигналов **sync\_in\_1** и **sync\_in\_2** имеются механизмы внутренней аппаратной генерации с заданной периодичностью. Разрешение генерации управляется параметрами **UEM\_SYNC\_IN\_1\_INTGEN** и **UEM\_SYNC\_IN\_2\_INTGEN** соответственно, а период повтора определяется параметрами **UEM\_IST1** и **UEM\_IST2** соответственно.

## Определения примитивных типов

Дескрипторы устройства и объекта, идентификатор параметра, логическое, 32- и 16-битные значения.

### Определения типов

* typedef ViSession **UEM\_DEVHANDLE**

*Дескриптор устройства УЭМ или виртуального устройства в составе УЭМ.*

* typedef ViSession **UEM\_OBJHANDLE**

*Дескриптор объекта в ОЗУ УЭМ.*

* typedef ViBoolean **UEM\_BOOL**

*Логическое значение.*

* typedef ViUInt16 **UEM\_PARAMID**

*Идентификатор параметра.*

* typedef ViUInt32 **UEM\_DWORD**

*32-битное целое без знака.*

* typedef ViUInt16 **UEM\_WORD**

*16-битное целое без знака.*

### Подробное описание

Дескрипторы устройства и объекта, идентификатор параметра, логическое, 32- и 16-битные значения.

### Типы

#### typedef ViSession UEM\_DEVHANDLE

Дескриптор устройства УЭМ или виртуального устройства в составе УЭМ.

#### typedef ViSession UEM\_OBJHANDLE

Дескриптор объекта в ОЗУ УЭМ.

#### typedef ViBoolean UEM\_BOOL

Логическое значение.

#### typedef ViUInt16 UEM\_PARAMID

Идентификатор параметра.

Возможные значения представлены в разделе **Описание параметров**.

#### typedef ViUInt32 UEM\_DWORD

32-битное целое без знака.

#### typedef ViUInt16 UEM\_WORD

16-битное целое без знака.

## Установление и разрыв связи с устройством

Описания функций установления и разрыва связи с устройством.

### Функции

* ViStatus **uem\_init** (ViRsrc idstr, ViBoolean idn, ViBoolean reset, ViSession \*uem)

*Инициализация объекта УЭМ.*

* ViStatus **uem\_connect** (ViSession uem, ViSession vi, ViUInt16 meznum, ViBoolean idn, ViBoolean reset)

*Привязка объекта УЭМ к сеансу носителя мезонина.*

* ViStatus **uem\_bc\_init** (**UEM\_DEVHANDLE** \*bc, **UEM\_DEVHANDLE** uem)

*Открытие виртуального КШ в составе УЭМ.*

* ViStatus **uem\_rt\_init** (**UEM\_DEVHANDLE** \*rt, **UEM\_DEVHANDLE** uem, **UEM\_WORD** rtaddr)

*Открытие виртуального ОУ в составе УЭМ.*

* ViStatus **uem\_bm\_init** (**UEM\_DEVHANDLE** \*bm, **UEM\_DEVHANDLE** uem)

*Открытие виртуального МШ в составе УЭМ.*

* ViStatus **uem\_close** (**UEM\_DEVHANDLE** anydev)

*Закрытие УЭМ или любого виртуального устройства в составе УЭМ.*

### Подробное описание

Описания функций установления и разрыва связи с устройством.

Приведенные здесь функции устанавливают связь с устройством УЭМ и с виртуальными устройствами (КШ/ОУ/МШ) в составе УЭМ. Рекомендуется также ознакомиться с разделами **Порядок действий при установлении связи с устройством** и **Виртуальные устройства**.

### Функции

#### ViStatus uem\_init (ViRsrc *idstr*, ViBoolean *idn*, ViBoolean *reset*, ViSession \* *uem*)

Инициализация объекта УЭМ.

Рекомендуется ознакомиться с подразделом **Порядок действий при установлении связи с устройством**.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *idstr* | В данном параметре передается **Адресная строка** VISA. Должна идентифицировать носитель мезонинов. |
| in | *idn* | Данный параметр определяет, производить ли идентификацию устройства в процедуре инициализации.  Допустимые значения:  VI\_OFF (0) - Не производить идентификацию.  VI\_ON (1) - Производить идентификацию (по умолчанию).  **Примечание:** Обычно не следует отключать идентификацию устройства, так как это дает дополнительную проверку на соответствие типа устройства тому, на который рассчитан драйвер. |
| in | *reset* | Данный параметр определяет, производить ли сброс устройства в процедуре инициализации.  Допустимые значения:  VI\_OFF (0) - Не производить сброс.  VI\_ON (1) - Производить сброс (по умолчанию). |
| out | *uem* | В данной переменной функция возвращает дескриптор устройства УЭМ, который необходимо сохранить для всех последующих вызовов функций драйвера устройства.  **Примечания:**  1) При каждом новом вызове функции инициализации открывается еще один сеанс связи с тем же устройством.  2) Обратите внимание, что типы UEM\_DEVHANDLE и ViSession являются синонимами. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_connect (ViSession *uem*, ViSession *vi*, ViUInt16 *meznum*, ViBoolean *idn*, ViBoolean *reset*)

Привязка объекта УЭМ к сеансу носителя мезонина.

Рекомендуется ознакомиться с подразделом **Порядок действий при установлении связи с устройством**.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *uem* | Дескриптор устройства, возвращенный функцией **uem\_init()**. |
| in | *vi* | Номер сеанса носителя мезонинов, возвращенный функцией unmbase\_init() [**4**]. |
| in | *meznum* | В данном параметре указывается номера позиции инструмента устройства УЭМ в носителе мезонинов. Допустимые значения 1-4. |
| in | *idn* | Данный параметр определяет, производить ли идентификацию устройства.  Допустимые значения:  VI\_OFF (0) - Не производить идентификацию.  VI\_ON (1) - Производить идентификацию (по умолчанию).  **Примечание:** Обычно не следует отключать идентификацию устройства, так как это дает дополнительную проверку на соответствие типа устройства тому, на который рассчитан драйвер. |
| in | *reset* | Данный параметр определяет, производить ли сброс устройства.  Допустимые значения:  VI\_OFF (0) - Не производить сброс.  VI\_ON (1) - Производить сброс (по умолчанию). |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_bc\_init (UEM\_DEVHANDLE \* *bc*, UEM\_DEVHANDLE *uem*)

Открытие виртуального КШ в составе УЭМ.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| out | *bc* | Дескриптор КШ. |
| in | *uem* | Дескриптор УЭМ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

При повторном открытии КШ будет возвращен дескриптор уже открытого КШ.

#### ViStatus uem\_rt\_init (UEM\_DEVHANDLE \* *rt*, UEM\_DEVHANDLE *uem*, UEM\_WORD *rtaddr*)

Открытие виртуального ОУ в составе УЭМ.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| out | *rt* | Дескриптор виртуального ОУ. |
| in | *uem* | Дескриптор УЭМ. |
| in | *rtaddr* | Адрес ОУ. Допустимые значения 0-30 (0-31 в сетях с запрещенными групповыми командами). |

##### Возвращает:

Код завершения. См. **Коды завершения**.

При повторном открытии ОУ с указанным номером будет возвращен дескриптор уже открытого ОУ с этим номером.

#### ViStatus uem\_bm\_init (UEM\_DEVHANDLE \* *bm*, UEM\_DEVHANDLE *uem*)

Открытие виртуального МШ в составе УЭМ.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| out | *bm* | Дескриптор МШ. |
| in | *uem* | Дескриптор УЭМ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

При повторном открытии МШ будет возвращен дескриптор уже открытого МШ.

#### ViStatus uem\_close (UEM\_DEVHANDLE *anydev*)

Закрытие УЭМ или любого виртуального устройства в составе УЭМ.

При закрытии виртуального устройства автоматически выполняется действие **uem\_stop()** для этого устройства, если необходимо, сброс виртуального устройства (**uem\_reset()**) и освобождение всех объектов в ОЗУ виртуального устройства. Все дескрипторы данного виртуального устройства после этого считаются недействительными. Остальные виртуальные устройства не затрагиваются.

При закрытии УЭМ в целом выполняется закрытие всех виртуальных устройств, после чего разрывается связь с устройством УЭМ. Дескриптор УЭМ после этого считается недействительным.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *anydev* | Дескриптор УЭМ или виртуального устройства (КШ, ОУ, МШ) в составе УЭМ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

## Действия с дескрипторами

Описания функций исследования дескрипторов объектов библиотеки и навигации между ними.

### Перечисления

* enum **UEM\_HANDLE\_TYPE** { **UEM\_INVH**, **UEM\_UEM**, **UEM\_BC**, **UEM\_RT**, **UEM\_BM**, **UEM\_CSEG**, **UEM\_BCP**, **UEM\_RESP** }

*Тип дескриптора объекта библиотеки УЭМ.*

### Функции

* **UEM\_HANDLE\_TYPE** **uem\_handle\_type** (**UEM\_DEVHANDLE** anyobject)

*Тип дескриптора объекта библиотеки УЭМ.*

* ViStatus **uem\_parent\_dev** (**UEM\_DEVHANDLE** anyobject, **UEM\_DEVHANDLE** \*parentdev)

*Родительское устройство.*

* ViStatus **uem\_root\_dev** (**UEM\_DEVHANDLE** anyobject, **UEM\_DEVHANDLE** \*uem)

*Физическое устройство УЭМ.*

* ViStatus **uem\_layer\_handle** (**UEM\_DEVHANDLE** uem, ViUInt32 sel, ViSession \*handle)

*Связь с ПО нижележащих слоев.*

### Селектор сеанса

* #define **UEM\_SEL\_UNMUEM**  1

*Сеанс низкоуровневого драйвера УЭМ unmuem.*

* #define **UEM\_SEL\_UNMBASE**  2

*Сеанc драйвера носителя мезонинов unmbase.*

* #define **UEM\_SEL\_UNBASE\_INT**  3

*"Внутренний" сеанс драйвера носителя мезонинов unmbase.*

### Подробное описание

Описания функций исследования дескрипторов объектов библиотеки и навигации между ними.

Приведенные здесь функции позволяют определить тип дескриптора и переходить по иерархии дескрипторов.

### Макросы

#### #define UEM\_SEL\_UNMUEM  1

Сеанс низкоуровневого драйвера УЭМ unmuem.

#### #define UEM\_SEL\_UNMBASE  2

Сеанc драйвера носителя мезонинов unmbase.

#### #define UEM\_SEL\_UNBASE\_INT  3

"Внутренний" сеанс драйвера носителя мезонинов unmbase.

### Перечисления

#### enum UEM\_HANDLE\_TYPE

Тип дескриптора объекта библиотеки УЭМ.

**Элементы перечислений:**

***UEM\_INVH*** Недействительный дескриптор.

***UEM\_UEM*** Физическое устройство УЭМ.

***UEM\_BC*** Виртуальный КШ.

***UEM\_RT*** Виртуальное ОУ.

***UEM\_BM*** Виртуальный МШ.

***UEM\_CSEG*** Командный сегмент.

***UEM\_BCP*** Программа КШ.

***UEM\_RESP*** Ответный сегмент.

### Функции

#### UEM\_HANDLE\_TYPE uem\_handle\_type (UEM\_DEVHANDLE *anyobject*)

Тип дескриптора объекта библиотеки УЭМ.

Возвращаемое значение позволяет определить тип объекта, дескриптор которого предъявлен.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *anyobject* | Дескриптор устройства или объекта в ОЗУ устройства. |

##### Возвращает:

Тип дескриптора.

#### ViStatus uem\_parent\_dev (UEM\_DEVHANDLE *anyobject*, UEM\_DEVHANDLE \* *parentdev*)

Родительское устройство.

Функция возвращает:

для объекта в ОЗУ УЭМ (допустимо указывать в аргументе anyobject) - дескриптор виртуального устройства, которому принадлежит данный объект,

для виртуального устройства - дескриптор УЭМ,

для УЭМ - дескриптор самого УЭМ.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *anyobject* | Дескриптор устройства или объекта в ОЗУ, родительское устройство которого требуется выяснить. |
| out | *parentdev* | В этом аргументе передается адрес переменной, в которую будет записан искомый дескриптор. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_root\_dev (UEM\_DEVHANDLE *anyobject*, UEM\_DEVHANDLE \* *uem*)

Физическое устройство УЭМ.

Для любого объекта в ОЗУ или виртуального устройства функция возвращает дескриптор УЭМ, которому принадлежит этот объект или виртуальное устройство.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *anyobject* | Дескриптор устройства или объекта в ОЗУ, по которому требуется определить дескриптор УЭМ. |
| out | *uem* | В этом аргументе передается адрес переменной, в которую будет записан искомый дескриптор УЭМ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_layer\_handle (UEM\_DEVHANDLE *uem*, ViUInt32 *sel*, ViSession \* *handle*)

Связь с ПО нижележащих слоев.

Настоящая расширенная библиотека полнофункциональна и позволяет решать все задачи по управлению устройствами УЭМ и их использованию без обращения к другим компонентам программного обеспечения.

(Естественным исключением является процедура инициализации, в ходе которой приложение обращается к функциям менеджера ресурсов VISA и к функции unmbase\_init() драйвера носителя мезонина [**4**].)

Тем не менее библиотека не препятствует работе с нижележащими слоями программного обеспечения. Расширенная библиотека uem работает поверх драйвера мезонина unmuem [**5**, **6**], который, в свою очередь, работает поверх драйвера носителя мезонинов unmbase [**4**].

Данная функция позволяет получить дескрипторы сеансов работы этих компонентов программного обеспечения и работать с ними напрямую.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *uem* | Дескриптор УЭМ. |
| in | *sel* | Данный аргумент определяет выбор нижележащего ПО и его дескриптора, см. константы **UEM\_SEL\_XXXXX**. |
| out | *handle* | В этой переменной возвращается запрошенный дескриптор сеанса. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

## Параметры конфигурации УЭМ

Описания параметров конфигурации и функций для работы с ними.

### Группы

* **Описание параметров**

*Описания параметров и определения констант - идентификаторов параметров конфигурации.*

* **Значения параметров**

### *Определения констант - значений параметров конфигурации.*

### Функции

* ViStatus **uem\_param\_get** (**UEM\_DEVHANDLE** uem, **UEM\_PARAMID** paramid, **UEM\_DWORD** \*value)

*Считывание значения конфигурационного параметра.*

* ViStatus **uem\_param\_set** (**UEM\_DEVHANDLE** uem, **UEM\_PARAMID** paramid, **UEM\_DWORD** value)

*Запись значения конфигурационного параметра.*

### Подробное описание

Описания параметров конфигурации и функций для работы с ними.

Под параметром конфигурации понимается отдельный бит или связанная по смыслу и расположению группа битов в регистрах УЭМ. Параметры конфигурации относятся к УЭМ в целом, а не к виртуальным устройствам в составе УЭМ. Каждому параметру конфигурации присвоен фиксированный числовой идентификатор, который передается в функции считывания и записи параметров.

### Функции

#### ViStatus uem\_param\_get (UEM\_DEVHANDLE *uem*, UEM\_PARAMID *paramid*, UEM\_DWORD \* *value*)

Считывание значения конфигурационного параметра.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *uem* | Дескриптор УЭМ.  Допускается указывать вместо дескриптора УЭМ дескриптор любого виртуального устройства в составе этого УЭМ. Функция выполняет такой вызов, как если бы был указан дескриптор УЭМ. |
| in | *paramid* | Идентификатор параметра. См. **Описание параметров**. |
| out | *value* | Значение параметра. См. **Значения параметров**. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_param\_set (UEM\_DEVHANDLE *uem*, UEM\_PARAMID *paramid*, UEM\_DWORD *value*)

Запись значения конфигурационного параметра.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *uem* | Дескриптор УЭМ.  Допускается указывать вместо дескриптора УЭМ дескриптор любого виртуального устройства в составе этого УЭМ. Функция выполняет такой вызов, как если бы был указан дескриптор УЭМ. |
| in | *paramid* | Идентификатор параметра. См. **Описание параметров**. |
| in | *value* | Значение параметра. См. **Значения параметров**. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

## Описание параметров

Описания параметров и определения констант - идентификаторов параметров конфигурации.

### Макросы

* #define **UEM\_TMTA\_DIS**  UEMi\_MAKE\_PARAMID ( 0, 2, 2)

*Запрет работы передатчика шины А.*

* #define **UEM\_TMTB\_DIS**  UEMi\_MAKE\_PARAMID ( 0, 3, 3)

*Запрет работы передатчика шины Б.*

* #define **UEM\_RCVA\_DIS**  UEMi\_MAKE\_PARAMID ( 0, 4, 4)

*Запрет работы приемника шины А.*

* #define **UEM\_RCVB\_DIS**  UEMi\_MAKE\_PARAMID ( 0, 5, 5)

*Запрет работы приемника шины Б.*

* #define **UEM\_SYNC\_IN\_1\_ENA**  UEMi\_MAKE\_PARAMID ( 0, 8, 8)

*Разрешение входной синхронизации 1.*

* #define **UEM\_SYNC\_IN\_2\_ENA**  UEMi\_MAKE\_PARAMID ( 0, 9, 9)

*Разрешение входной синхронизации 2.*

* #define **UEM\_SYNC\_OUT\_1\_ENA**  UEMi\_MAKE\_PARAMID ( 0, 10, 10)

*Разрешение выходной синхронизации 1.*

* #define **UEM\_SYNC\_OUT\_2\_ENA**  UEMi\_MAKE\_PARAMID ( 0, 11, 11)

*Разрешение выходной синхронизации 2.*

* #define **UEM\_BRCST\_DIS**  UEMi\_MAKE\_PARAMID ( 0, 12, 12)

*Запрет групповых сообщений.*

* #define **UEM\_SYNC\_IN\_1\_INTGEN**  UEMi\_MAKE\_PARAMID( 0, 13, 13)

*Разрешение внутренней эмуляции сигнала входной синхронизации 1.*

* #define **UEM\_ERR\_INJ\_DIS**  UEMi\_MAKE\_PARAMID ( 0, 14, 14)

*Запрет внесения ошибок кодирования в передаваемую в МКПД информацию для КШ и ОУ.*

* #define **UEM\_SYNC\_IN\_2\_INTGEN**  UEMi\_MAKE\_PARAMID( 0, 29, 29)

*Разрешение внутренней эмуляции сигнала входной синхронизации 2.*

* #define **UEM\_TMT\_RES**  UEMi\_MAKE\_PARAMID ( 5, 4, 4)

*Сброс настроек передатчиков (только запись).*

* #define **UEM\_SYNC\_IN\_1\_SET**  UEMi\_MAKE\_PARAMID ( 5, 10, 10)

*Программная генерация сигнала входной синхронизации 1 (только запись).*

* #define **UEM\_SYNC\_IN\_2\_SET**  UEMi\_MAKE\_PARAMID ( 5, 11, 11)

*Программная генерация сигнала входной синхронизации 2 (только запись).*

* #define **UEM\_SYNC\_OUT\_1\_SET**  UEMi\_MAKE\_PARAMID ( 5, 12, 12)

*Программная генерация сигнала выходной синхронизации 1 (только запись).*

* #define **UEM\_SYNC\_OUT\_2\_SET**  UEMi\_MAKE\_PARAMID ( 5, 13, 13)

*Программная генерация сигнала выходной синхронизации 2 (только запись).*

* #define **UEM\_DB\_ACT**  UEMi\_MAKE\_PARAMID ( 5, 31, 31)

*Обнаружена передача данных по шине (только чтение).*

* #define **UEM\_TXA\_RFT**  UEMi\_MAKE\_PARAMID ( 6, 31, 16)

*Управление длительностью фронта и среза при передаче в шину А.*

* #define **UEM\_TXA\_VPP**  UEMi\_MAKE\_PARAMID ( 6, 15, 0)

*Управление размахом сигнала при передаче в шину А.*

* #define **UEM\_TXB\_RFT**  UEMi\_MAKE\_PARAMID ( 7, 31, 16)

*Управление длительностью фронта и среза при передаче в шину Б.*

* #define **UEM\_TXB\_VPP**  UEMi\_MAKE\_PARAMID ( 7, 15, 0)

*Управление размахом сигнала при передаче в шину Б.*

* #define **UEM\_MC\_DIS**  UEMi\_MAKE\_PARAMID (0x0C, 0, 0)

*Запрет команд управления.*

* #define **UEM\_BTMT\_DIS**  UEMi\_MAKE\_PARAMID (0x0C, 1, 1)

*Запрет блокирования и разблокирования передатчиков ОУ.*

* #define **UEM\_BRTF\_DIS**  UEMi\_MAKE\_PARAMID (0x0C, 2, 2)

*Запрет блокирования и разблокирования признака неисправности ОУ.*

* #define **UEM\_SYNC2\_VRTA**  UEMi\_MAKE\_PARAMID (0x0C, 7, 3)

*Номер ОУ – условие выработки сигнала выходной синхронизации 2.*

* #define **UEM\_SYNC1\_C\_D\_**  UEMi\_MAKE\_PARAMID (0x0C, 8, 8)

*Командное/ответное (1) слово или слово данных (0) – условие выработки сигнала выходной синхронизации 1.*

* #define **UEM\_SYNC1\_ERR**  UEMi\_MAKE\_PARAMID (0x0C, 9, 9)

*Наличие ошибок в слове – условие выработки сигнала выходной синхронизации 1.*

* #define **UEM\_SYNC1\_GAPB**  UEMi\_MAKE\_PARAMID (0x0C, 10, 10)

*Наличие паузы перед словом - условие выработки сигнала выходной синхронизации 1.*

* #define **UEM\_SYNC1\_CH**  UEMi\_MAKE\_PARAMID (0x0C, 12, 11)

*Слово передается по указанной шине - условие выработки сигнала выходной синхронизации 1.*

* #define **UEM\_IST2**  UEMi\_MAKE\_PARAMID (0x0C, 31, 16)

*Период внутренней генерации сигнала входной синхронизации 2.*

* #define **UEM\_IST1**  UEMi\_MAKE\_PARAMID ( 8, 31, 16)

*Период внутренней генерации сигнала входной синхронизации 1.*

* #define **UEM\_BM\_WORD\_PATTERN**  UEMi\_MAKE\_PARAMID (0x0F, 15, 0)

*Шаблон (значение) слова - условие выработки сигнала выходной синхронизации 1.*

* #define **UEM\_BM\_WORD\_MASK**  UEMi\_MAKE\_PARAMID (0x0F, 31, 16)

*Маска побитного сравнения слова с шаблоном - условие выработки сигнала выходной синхронизации 1.*

### Подробное описание

Описания параметров и определения констант - идентификаторов параметров конфигурации.

В этом разделе перечислены константы, идентифицирующие параметры конфигурации УЭМ. Они предназначены для использования в качестве значения аргумента **paramid** в функциях **uem\_param\_set()**, **uem\_param\_get()**.

Для каждого идентификатора параметра приведено описание назначения этого параметра.

Могут быть заданы следующие конфигурационные параметры УЭМ:

* Параметры, задающие включение и отключение шин А и Б.
* Параметры, задающие правила функционирования КШ, ОУ и МШ.
* Параметры, управляющие внесением ошибок в передаваемую в МКПД информацию.
* Параметры, управляющие всеми видами синхронизации.
* Электрические параметры передатчиков.

### Макросы

#### #define UEM\_TMTA\_DIS  UEMi\_MAKE\_PARAMID ( 0, 2, 2)

Запрет работы передатчика шины А.

Установленный флаг (UEM\_TMTA\_DIS=1) запрещает работу передатчика шины А (основной магистрали МКПД).

В этом случае прием передаваемой информации по данной шине виртуальными устройствами производится через внутренние связи (режим «off-line»).

См. **Значения параметра UEM\_TMTA\_DIS**.

#### #define UEM\_TMTB\_DIS  UEMi\_MAKE\_PARAMID ( 0, 3, 3)

Запрет работы передатчика шины Б.

Установленный флаг (UEM\_TMTB\_DIS=1) запрещает работу передатчика шины Б (резервной магистрали МКПД).

В этом случае прием передаваемой информации по данной шине виртуальными устройствами производится через внутренние связи (режим «off-line»). См. **Значения параметра UEM\_TMTB\_DIS**.

#### #define UEM\_RCVA\_DIS  UEMi\_MAKE\_PARAMID ( 0, 4, 4)

Запрет работы приемника шины А.

Установленный флаг (UEM\_RCVA\_DIS=1) запрещает работу приемника шины А (основной магистрали МКПД).

В этом случае не анализируется состояние входных сигналов управления МКПД, подключаемых к шине А. Прием передаваемой информации по данной шине виртуальными устройствами производится через внутренние связи (режим «off-line»).

См. **Значения параметра UEM\_RCVA\_DIS**.

#### #define UEM\_RCVB\_DIS  UEMi\_MAKE\_PARAMID ( 0, 5, 5)

Запрет работы приемника шины Б.

Установленный флаг (UEM\_RCVB\_DIS=1) запрещает работу приемника шины Б (резервной магистрали МКПД).

В этом случае не анализируется состояние входных сигналов управления МКПД, подключаемых к шине Б. Прием передаваемой информации по данной шине виртуальными устройствами производится через внутренние связи (режим «off-line»).

См. **Значения параметра UEM\_RCVB\_DIS**.

#### #define UEM\_SYNC\_IN\_1\_ENA  UEMi\_MAKE\_PARAMID ( 0, 8, 8)

Разрешение входной синхронизации 1.

Установленный флаг (UEM\_SYNC\_IN\_1\_ENA=1) разрешает обработку входного сигнала синхронизации 1. Сигнал входной синхронизации 1 (**sync\_in\_1**) обеспечивает синхронную с внешней аппаратурой пересылку в МКПД слов для КШ.

См. **Управление синхронизацией**.

См. **Значения параметра UEM\_SYNC\_IN\_1\_ENA**.

#### #define UEM\_SYNC\_IN\_2\_ENA  UEMi\_MAKE\_PARAMID ( 0, 9, 9)

Разрешение входной синхронизации 2.

Установленный флаг (UEM\_SYNC\_IN\_2\_ENA=1) разрешает обработку входного сигнала синхронизации 2. Входной сигнал синхронизации 2 (**sync\_in\_2**) обеспечивает запуск в работу КШ, предварительно сконфигурированного с требуемыми значениями параметров, а также – повторный запуск.

См. **Управление синхронизацией**.

См. **Значения параметра UEM\_SYNC\_IN\_2\_ENA**.

#### #define UEM\_SYNC\_OUT\_1\_ENA  UEMi\_MAKE\_PARAMID ( 0, 10, 10)

Разрешение выходной синхронизации 1.

Установленный флаг (UEM\_SYNC\_OUT\_1\_ENA=1) разрешает выработку сигнала выходной синхронизации 1. Сигнал выходной синхронизации 1 (**sync\_out\_1**), если разрешен, формируется, когда МШ обнаружил слово в МКПД с заданным значением разрядов и заданными характеристиками. Характеристики, определяющие условия выработки сигнала выходной синхронизации 1, задаются параметрами **UEM\_SYNC1\_C\_D\_**, **UEM\_SYNC1\_ERR**, **UEM\_SYNC1\_GAPB**, **UEM\_SYNC1\_CH**.

См. **Управление синхронизацией**.

См. **Значения параметра UEM\_SYNC\_OUT\_1\_ENA**.

#### #define UEM\_SYNC\_OUT\_2\_ENA  UEMi\_MAKE\_PARAMID ( 0, 11, 11)

Разрешение выходной синхронизации 2.

Установленный флаг (UEM\_SYNC\_OUT\_2\_ENA=1) разрешает выработку сигнала выходной синхронизации 2. Сигнал выходной синхронизации 2 (**sync\_out\_2**), если разрешен, вырабатывается, когда МШ обнаружил достоверное командное (ответное) слово с заданным адресом ОУ. Адрес ОУ задается параметром **UEM\_SYNC2\_VRTA**.

См. **Управление синхронизацией**.

См. **Значения параметра UEM\_SYNC\_OUT\_2\_ENA**.

#### #define UEM\_BRCST\_DIS  UEMi\_MAKE\_PARAMID ( 0, 12, 12)

Запрет групповых сообщений.

Установленный флаг (UEM\_BRCST\_DIS=1) запрещает групповые сообщения.

В этом случае ОУ с адресом 31 (если данный адрес был активизирован) используется в качестве «обычного», «не широковещательного» адреса ОУ, используемого для приема и передачи информации.

В аппаратуре этот флаг влияет на подсистему ОУ, в расширенной библиотеке - на работу виртуальных устройств всех типов.

См. **Значения параметра UEM\_BRCST\_DIS**.

#### #define UEM\_SYNC\_IN\_1\_INTGEN  UEMi\_MAKE\_PARAMID( 0, 13, 13)

Разрешение внутренней эмуляции сигнала входной синхронизации 1.

Установленный флаг (UEM\_SYNC\_IN\_1\_INTGEN=1) разрешает внутреннюю эмуляцию сигнала входной синхронизации 1.

См. **Управление синхронизацией**.

См. **Значения параметра UEM\_SYNC\_IN\_1\_INTGEN**.

#### #define UEM\_ERR\_INJ\_DIS  UEMi\_MAKE\_PARAMID ( 0, 14, 14)

Запрет внесения ошибок кодирования в передаваемую в МКПД информацию для КШ и ОУ.

Установленный флаг (UEM\_ERR\_INJ\_DIS=1) запрещает внесения ошибок кодирования в передаваемую в МКПД информацию для КШ и ОУ.

В этом случае информация о внесении ошибок в командные и ответные сегменты сообщений будет игнорироваться.

См. **Значения параметра UEM\_ERR\_INJ\_DIS**.

#### #define UEM\_SYNC\_IN\_2\_INTGEN  UEMi\_MAKE\_PARAMID( 0, 29, 29)

Разрешение внутренней эмуляции сигнала входной синхронизации 2.

Установленный флаг (UEM\_SYNC\_IN\_2\_INTGEN=1) разрешает внутреннюю эмуляцию сигнала входной синхронизации 2.

См. **Управление синхронизацией**.

См. **Значения параметра UEM\_SYNC\_IN\_2\_INTGEN**.

#### #define UEM\_TMT\_RES  UEMi\_MAKE\_PARAMID ( 5, 4, 4)

Сброс настроек передатчиков (только запись).

Параметр доступен только по записи.

При установлении данного параметра конфигурации в 1 осуществляется приведение в исходное состояние настроек по регулировке параметров выходных сигналов передатчиков и установка значений по умолчанию в параметрах **UEM\_TXA\_RFT**, **UEM\_TXA\_VPP**, **UEM\_TXB\_RFT**, **UEM\_TXB\_VPP**.

См. **Значение UEM\_SET**.

#### #define UEM\_SYNC\_IN\_1\_SET  UEMi\_MAKE\_PARAMID ( 5, 10, 10)

Программная генерация сигнала входной синхронизации 1 (только запись).

Флаг доступен только по записи.

Установка данного флага (UEM\_SYNC\_IN\_1\_SET=1) является командой программной имитации поступления входного сигнала внешней синхронизации 1.

См. **Управление синхронизацией**.

См. **Значение UEM\_SET**.

#### #define UEM\_SYNC\_IN\_2\_SET  UEMi\_MAKE\_PARAMID ( 5, 11, 11)

Программная генерация сигнала входной синхронизации 2 (только запись).

Флаг доступен только по записи.

Установка данного флага (UEM\_SYNC\_IN\_2\_SET=1) является командой программной имитации поступления входного сигнала внешней синхронизации 2.

См. **Управление синхронизацией**.

См. **Значение UEM\_SET**.

#### #define UEM\_SYNC\_OUT\_1\_SET  UEMi\_MAKE\_PARAMID ( 5, 12, 12)

Программная генерация сигнала выходной синхронизации 1 (только запись).

Флаг доступен только по записи.

Установка данного флага (UEM\_SYNC\_OUT\_1\_SET=1) является командой программной имитации выработки выходного сигнала внешней синхронизации 1.

См. **Управление синхронизацией**.

См. **Значение UEM\_SET**.

#### #define UEM\_SYNC\_OUT\_2\_SET  UEMi\_MAKE\_PARAMID ( 5, 13, 13)

Программная генерация сигнала выходной синхронизации 2 (только запись).

Флаг доступен только по записи.

Установка данного флага (UEM\_SYNC\_OUT\_2\_SET=1) является командой программной имитации выработки выходного сигнала внешней синхронизации 2.

См. **Управление синхронизацией**.

См. **Значение UEM\_SET**.

#### #define UEM\_DB\_ACT  UEMi\_MAKE\_PARAMID ( 5, 31, 31)

Обнаружена передача данных по шине (только чтение).

Флаг доступен только по чтению.

Установленный флаг (UEM\_DB\_ACT=1) указывает на текущую активность в любой из шин МКПД по информации декодеров, с «инерционностью» 0,5с от момента обнаружения отсутствия активности по обеим шинам МКПД.

См. **Управление синхронизацией**.

#### #define UEM\_TXA\_RFT  UEMi\_MAKE\_PARAMID ( 6, 31, 16)

Управление длительностью фронта и среза при передаче в шину А.

Параметр задает длительность фронта (среза) импульсов выходного сигнала передатчика МКПД для шины А.

В **таблице 1** в разделе **Параметры передатчиков и характеристики выходных сигналов** приведена приблизительная оценка соответствия значений данного параметра и значений сигнала МКПД.

См. **Значения параметров UEM\_TXA\_RFT, UEM\_TXB\_RFT**.

#### #define UEM\_TXA\_VPP  UEMi\_MAKE\_PARAMID ( 6, 15, 0)

Управление размахом сигнала при передаче в шину А.

Параметр задает размах выходного сигнала передатчика МКПД для шины А. Допустимые значения 0x000E – 0x01FF. Значение по умолчанию 0x0136.

В **таблице 2** в разделе **Параметры передатчиков и характеристики выходных сигналов** приведена приблизительная оценка соответствия значений параметров и значений сигналов МКПД при условии работы на номинальную эквивалентную нагрузку по ГОСТ Р 52070.

См. **Значения параметров UEM\_TXA\_VPP, UEM\_TXB\_VPP**.

#### #define UEM\_TXB\_RFT  UEMi\_MAKE\_PARAMID ( 7, 31, 16)

Управление длительностью фронта и среза при передаче в шину Б.

Параметр, задающий длительность фронта (среза) импульсов выходного сигнала передатчика МКПД для шины Б.

В **таблице 1** в разделе **Параметры передатчиков и характеристики выходных сигналов** приведена приблизительная оценка соответствия значений данного параметра и значений сигнала МКПД.

См. **Значения параметров UEM\_TXA\_RFT, UEM\_TXB\_RFT**.

#### #define UEM\_TXB\_VPP  UEMi\_MAKE\_PARAMID ( 7, 15, 0)

Управление размахом сигнала при передаче в шину Б.

Параметр, задающий размах выходного сигнала передатчика МКПД для шины Б. Разряды [15:9] доступны по записи и по чтению. Значения по умолчанию 0136h.

В **таблице 2** в разделе **Параметры передатчиков и характеристики выходных сигналов** приведена приблизительная оценка соответствия значений данного параметра и значений сигнала МКПД при условии работы на номинальную эквивалентную нагрузку по ГОСТ Р 52070.

См. **Значения параметров UEM\_TXA\_VPP, UEM\_TXB\_VPP**.

#### #define UEM\_MC\_DIS  UEMi\_MAKE\_PARAMID (0x0C, 0, 0)

Запрет команд управления.

Установленный флаг (UEM\_MC\_DIS=1) является признаком «запрета» команд управления, т.е КС с кодами подадресов 00000 и 11111 будут интерпретироваться активными ОУ как команды информационного обмена. В аппаратуре этот флаг влияет на подсистему ОУ, в расширенной библиотеке - на работу виртуальных устройств всех типов.

См. **Значения параметра UEM\_MC\_DIS**.

#### #define UEM\_BTMT\_DIS  UEMi\_MAKE\_PARAMID (0x0C, 1, 1)

Запрет блокирования и разблокирования передатчиков ОУ.

Установленный флаг (UEM\_BTMT\_DIS=1) является признаком запрета для ОУ блокировок и разблокировок передатчиков при поступлении достоверных КУ «Блокировать передатчик», «Разблокировать передатчик», «Установить ОУ в исходное состояние», в том числе в групповых сообщениях (по умолчанию, при нулевом значении флага, такие блокировки и разблокировки осуществляются индивидуально для каждого адреса ОУ с использованием внутренних сигналов, запрещающих передачу информации по соответствующей шине (шинам)). Имеет смысл только при сброшенном флаге **UEM\_MC\_DIS**.

См. **Значения параметра UEM\_BTMT\_DIS**.

#### #define UEM\_BRTF\_DIS  UEMi\_MAKE\_PARAMID (0x0C, 2, 2)

Запрет блокирования и разблокирования признака неисправности ОУ.

Установленный флаг (UEM\_BRTF\_DIS=1) является признаком запрета для ОУ блокировок и разблокировок признака «Неисправность ОУ» в ОС при поступлении достоверных КУ «Блокировать признак неисправности ОУ», «Разблокировать признак неисправности ОУ», «Установить ОУ в исходное состояние», в том числе в групповых сообщениях (по умолчанию, при нулевом значении флага такие блокировки и разблокировки осуществляются индивидуально для каждого адреса ОУ с использованием внутренних сигналов, блокирующих установку «1» в соответствующем разряде ОС, исключая режим прямого формирования признаков ОС). Имеет смысл только при сброшенном флаге **UEM\_MC\_DIS**.

См. **Значения параметра UEM\_BRTF\_DIS**.

#### #define UEM\_SYNC2\_VRTA  UEMi\_MAKE\_PARAMID (0x0C, 7, 3)

Номер ОУ – условие выработки сигнала выходной синхронизации 2.

Параметр, задающий адрес ОУ, при обнаружении которого в достоверном КС (ОС) будет выработан выходной сигнал внешней синхронизации 2. Сигнал вырабатывается, если его выработка разрешена.

См. **Управление синхронизацией**.

См. **Значения параметра UEM\_SYNC2\_VRTA**.

#### #define UEM\_SYNC1\_C\_D\_  UEMi\_MAKE\_PARAMID (0x0C, 8, 8)

Командное/ответное (1) слово или слово данных (0) – условие выработки сигнала выходной синхронизации 1.

Параметр, задающий тип синхроимпульса слова, при обнаружении которого может быть выработан выходной сигнал синхронизации 1 – командное (ответное) (если UEM\_SYNC1\_C\_D\_=1) или слово данных (если UEM\_SYNC1\_C\_D\_=0). Сигнал вырабатывается, если его выработка разрешена, а также выполнены условия выработки, заданные другими параметрами.

См. **Управление синхронизацией**.

См. **Значения параметра UEM\_SYNC1\_C\_D\_**.

#### #define UEM\_SYNC1\_ERR  UEMi\_MAKE\_PARAMID (0x0C, 9, 9)

Наличие ошибок в слове – условие выработки сигнала выходной синхронизации 1.

Параметр, задающий условие наличия или отсутствия ошибки в слове, при обнаружении которого может быть выработан выходной сигнал синхронизации 1 – при условии наличия ошибки (если значение установлено в «1») или при условии отсутствия ошибки (если значение установлено в «0»). Сигнал вырабатывается, если его выработка разрешена, а также выполнены условия выработки, заданные другими параметрами.

См. **Управление синхронизацией**.

См. **Значения параметра UEM\_SYNC1\_ERR**.

#### #define UEM\_SYNC1\_GAPB  UEMi\_MAKE\_PARAMID (0x0C, 10, 10)

Наличие паузы перед словом - условие выработки сигнала выходной синхронизации 1.

Параметр, при единичном значении которого выходной сигнал внешней синхронизации 1 может вырабатываться только при наличии паузы перед заданным словом, при нулевом значении – только для слова, имеющего соприкосновение с предыдущим словом. Сигнал вырабатывается, если его выработка разрешена, а также выполнены условия выработки, заданные другими параметрами.

См. **Управление синхронизацией**.

См. **Значения параметра UEM\_SYNC1\_GAPB**.

#### #define UEM\_SYNC1\_CH  UEMi\_MAKE\_PARAMID (0x0C, 12, 11)

Слово передается по указанной шине - условие выработки сигнала выходной синхронизации 1.

Данный параметр определяет, будет ли сигнал выходной синхронизации 1 вырабатываться при обнаружении слова в шине А, или в шине Б, или независимо от шины, в которой слово обнаружено. Сигнал вырабатывается, если его выработка разрешена, а также выполнены условия выработки, заданные другими параметрами.

Допустимые значения: **UEM\_SYNC1\_CH\_A**, **UEM\_SYNC1\_CH\_B**, **UEM\_SYNC1\_ACH**.

См. **Управление синхронизацией**.

#### #define UEM\_IST2  UEMi\_MAKE\_PARAMID (0x0C, 31, 16)

Период внутренней генерации сигнала входной синхронизации 2.

В данном параметре задается период повторения сигнала входной синхронизации 2 при внутренней эмуляции, с разрешением 100 мкс на МЗР.

См. **Управление синхронизацией**.

См. **Значения параметров UEM\_IST1, UEM\_IST2**.

Нулевое значение параметра интерпретируется аппаратурой как 0x10000 (6,5536 с).

#### #define UEM\_IST1  UEMi\_MAKE\_PARAMID ( 8, 31, 16)

Период внутренней генерации сигнала входной синхронизации 1.

В данном параметре задается период повторения сигнала входной синхронизации 1 при внутренней эмуляции с разрешением 100 мкс на МЗР.

См. **Управление синхронизацией**.

См. **Значения параметров UEM\_IST1, UEM\_IST2**.

Нулевое значение параметра интерпретируется аппаратурой как 0x10000 (6,5536 с).

#### #define UEM\_BM\_WORD\_PATTERN  UEMi\_MAKE\_PARAMID (0x0F, 15, 0)

Шаблон (значение) слова - условие выработки сигнала выходной синхронизации 1.

В данном параметре задаются значения битов слова, при обнаружении которых может быть выработан сигнал выходной синхронизации 1. На совпадение проверяются только биты, для которых биты в соответствующих позициях параметра **UEM\_BM\_WORD\_MASK** (маски) установлены в 1. Сигнал вырабатывается, если его выработка разрешена, а также выполнены условия выработки, заданные другими параметрами.

См. **Управление синхронизацией**.

См. **Значения параметров UEM\_BM\_WORD\_PATTERN, UEM\_BM\_WORD\_MASK**.

#### #define UEM\_BM\_WORD\_MASK  UEMi\_MAKE\_PARAMID (0x0F, 31, 16)

Маска побитного сравнения слова с шаблоном - условие выработки сигнала выходной синхронизации 1.

Используется при сравнении обнаруженного слова МКПД с эталоном.

См. **UEM\_BM\_WORD\_PATTERN**.

См. **Управление синхронизацией**.

См. **Значения параметров UEM\_BM\_WORD\_PATTERN, UEM\_BM\_WORD\_MASK**.

## Значения параметров

Определения констант - значений параметров конфигурации.

### Значения параметра UEM\_TMTA\_DIS

* #define **UEM\_TMTA\_DISABLED**  1

*Передатчик шины А отключен.*

* #define **UEM\_TMTA\_ENABLED**  0

*Передатчик шины А включен.*

* #define **UEM\_TMTA\_DEFAULT**  (**UEM\_TMTA\_ENABLED**)

*По умолчанию: Передатчик шины А включен.*

### Значения параметра UEM\_TMTB\_DIS

* #define **UEM\_TMTB\_DISABLED**  1

*Передатчик шины Б отключен.*

* #define **UEM\_TMTB\_ENABLED**  0

*Передатчик шины Б включен.*

* #define **UEM\_TMTB\_DEFAULT**  (**UEM\_TMTB\_ENABLED**)

*По умолчанию: Передатчик шины Б включен.*

### Значения параметра UEM\_RCVA\_DIS

* #define **UEM\_RCVA\_DISABLED**  1

*Приемник шины А отключен.*

* #define **UEM\_RCVA\_ENABLED**  0

*Приемник шины А включен.*

* #define **UEM\_RCVA\_DEFAULT**  (**UEM\_RCVA\_ENABLED**)

*По умолчанию: Приемник шины А включен.*

### Значения параметра UEM\_RCVB\_DIS

* #define **UEM\_RCVB\_DISABLED**  1

*Приемник шины Б отключен.*

* #define **UEM\_RCVB\_ENABLED**  0

*Приемник шины Б включен.*

* #define **UEM\_RCVB\_DEFAULT**  (**UEM\_RCVB\_ENABLED**)

*По умолчанию: Приемник шины Б включен.*

### Значения параметра UEM\_SYNC\_IN\_1\_ENA

* #define **UEM\_SYNC\_IN\_1\_DISABLED**  0

*Обработка входного сигнала синхронизации 1 запрещена.*

* #define **UEM\_SYNC\_IN\_1\_ENABLED**  1

*Обработка входного сигнала синхронизации 1 разрешена.*

* #define **UEM\_SYNC\_IN\_1\_DEFAULT**  (**UEM\_SYNC\_IN\_1\_DISABLED**)

*По умолчанию: Обработка входного сигнала синхронизации 1 запрещена.*

### Значения параметра UEM\_SYNC\_IN\_2\_ENA

* #define **UEM\_SYNC\_IN\_2\_DISABLED**  0

*Обработка входного сигнала синхронизации 2 запрещена.*

* #define **UEM\_SYNC\_IN\_2\_ENABLED**  1

*Обработка входного сигнала синхронизации 2 разрешена.*

* #define **UEM\_SYNC\_IN\_2\_DEFAULT**  (**UEM\_SYNC\_IN\_2\_DISABLED**)

*По умолчанию: Обработка входного сигнала синхронизации 2 запрещена.*

### Значения параметра UEM\_SYNC\_OUT\_1\_ENA

* #define **UEM\_SYNC\_OUT\_1\_DISABLED**  0

*Формирование выходного сигнала синхронизации 1 запрещено.*

* #define **UEM\_SYNC\_OUT\_1\_ENABLED**  1

*Формирование выходного сигнала синхронизации 1 разрешено.*

* #define **UEM\_SYNC\_OUT\_1\_DEFAULT**  (**UEM\_SYNC\_OUT\_1\_DISABLED**)

*По умолчанию: Формирование выходного сигнала синхронизации 1 запрещено.*

### Значения параметра UEM\_SYNC\_OUT\_2\_ENA

* #define **UEM\_SYNC\_OUT\_2\_DISABLED**  0

*Формирование выходного сигнала синхронизации 2 запрещено.*

* #define **UEM\_SYNC\_OUT\_2\_ENABLED**  1

*Формирование выходного сигнала синхронизации 2 разрешено.*

* #define **UEM\_SYNC\_OUT\_2\_DEFAULT**  (**UEM\_SYNC\_OUT\_2\_DISABLED**)

*По умолчанию: Формирование выходного сигнала синхронизации 2 запрещено.*

### Значения параметра UEM\_BRCST\_DIS

* #define **UEM\_BRCST\_DISABLED**  1

*Групповые сообщения запрещены.*

* #define **UEM\_BRCST\_ENABLED**  0

*Групповые сообщения разрешены.*

* #define **UEM\_BRCST\_DEFAULT**  (**UEM\_BRCST\_ENABLED**)

*По умолчанию: Групповые сообщения разрешены.*

### Значения параметра UEM\_SYNC\_IN\_1\_INTGEN

* #define **UEM\_SYNC\_IN\_1\_INTGEN\_DISABLED**  0

*Внутренняя генерация запрещена.*

* #define **UEM\_SYNC\_IN\_1\_INTGEN\_ENABLED**  1

*Внутренняя генерация разрешена.*

* #define **UEM\_SYNC\_IN\_1\_INTGEN\_DEFAULT**  (**UEM\_SYNC\_IN\_1\_INTGEN\_DISABLED**)

*По умолчанию: запрещена.*

### Значения параметра UEM\_SYNC\_IN\_2\_INTGEN

* #define **UEM\_SYNC\_IN\_2\_INTGEN\_DISABLED**  0

*Внутренняя генерация запрещена.*

* #define **UEM\_SYNC\_IN\_2\_INTGEN\_ENABLED**  1

*Внутренняя генерация разрешена.*

* #define **UEM\_SYNC\_IN\_2\_INTGEN\_DEFAULT**  (**UEM\_SYNC\_IN\_2\_INTGEN\_DISABLED**)

*По умолчанию: запрещена.*

### Значения параметра UEM\_ERR\_INJ\_DIS

* #define **UEM\_ERR\_INJ\_DISABLED**  1

*Внесение ошибок кодирования запрещено.*

* #define **UEM\_ERR\_INJ\_ENABLED**  0

*Внесение ошибок кодирования разрешено.*

* #define **UEM\_ERR\_INJ\_DEFAULT**  (**UEM\_ERR\_INJ\_ENABLED**)

*По умолчанию: Внесение ошибок кодирования разрешено.*

### Значение UEM\_SET

Значение **UEM\_SET**, тождественно равное 1, используется с параметрами, фактически являющимися командами, вызывающими определенные действия аппаратуры УЭМ. Такие параметры доступны только по записи, причем действенной является именно запись значения 1. Такие параметры не имеют и не требуют установки значений по умолчанию.

* #define **UEM\_SET**  1

*Установка параметра в 1.*

### Значения параметра UEM\_DB\_ACT

Данный параметр доступен только по чтению, он не имеет и не требует установки значения по умолчанию.

* #define **UEM\_DB\_INACTIVE**  0

*Нет активности на шине данных.*

* #define **UEM\_DB\_ACTIVE**  1

*Обнаружена активность на шине данных.*

### Значения параметров UEM\_TXA\_RFT, UEM\_TXB\_RFT

* #define **UEM\_RFT\_MIN**  0

*Минимальное значение.*

* #define **UEM\_RFT\_MAX**  31

*Максимальное значение.*

* #define **UEM\_RFT\_SINE**  32

*Специальное значение. Включает формирование синусоидального сигнала.*

* #define **UEM\_RFT\_DEFAULT**  18

*Значение по умолчанию (18).*

### Значения параметров UEM\_TXA\_VPP, UEM\_TXB\_VPP

* #define **UEM\_VPP\_MIN**  0x00E

*Минимальное значение.*

* #define **UEM\_VPP\_MAX**  0x01FF

*Максимальное значение.*

* #define **UEM\_VPP\_DEFAULT**  0x0136

*Значение по умолчанию (0x0136).*

### Значения параметра UEM\_MC\_DIS

* #define **UEM\_MC\_DISABLED**  1

*Команды управления запрещены.*

* #define **UEM\_MC\_ENABLED**  0

*Команды управления разрешены.*

* #define **UEM\_MC\_DEFAULT**  (**UEM\_MC\_ENABLED**)

*По умолчанию: Команды управления разрешены.*

### Значения параметра UEM\_BTMT\_DIS

* #define **UEM\_BTMT\_DISABLED**  1

*Блокирование и разблокирование передатчиков ОУ запрещено.*

* #define **UEM\_BTMT\_ENABLED**  0

*Блокирование и разблокирование передатчиков ОУ разрешено.*

* #define **UEM\_BTMT\_DEFAULT**  (**UEM\_BTMT\_ENABLED**)

*По умолчанию: Блокирование и разблокирование передатчиков ОУ разрешено.*

### Значения параметра UEM\_BRTF\_DIS

* #define **UEM\_BRTF\_DISABLED**  1

*Блокирование и разблокирование признака неисправности ОУ запрещено.*

* #define **UEM\_BRTF\_ENABLED**  0

*Блокирование и разблокирование признака неисправности ОУ разрешено.*

* #define **UEM\_BRTF\_DEFAULT**  (**UEM\_BRTF\_ENABLED**)

*По умолчанию: Блокирование и разблокирование признака неисправности ОУ разрешено.*

### Значения параметра UEM\_SYNC2\_VRTA

* #define **UEM\_SYNC2\_VRTA\_MIN**  0

*Минимальное значение.*

* #define **UEM\_SYNC2\_VRTA\_MAX**  31

*Максимальное значение.*

* #define **UEM\_SYNC2\_VRTA\_DEFAULT**  0

*Значение по умолчанию.*

### Значения параметра UEM\_SYNC1\_C\_D\_

* #define **UEM\_SYNC1\_ON\_COMMAND**  1

*Синхроимпульс командного/ответного слова.*

* #define **UEM\_SYNC1\_ON\_DATA**  0

*Синхроимпульс слова данных.*

* #define **UEM\_SYNC1\_C\_D\_DEFAULT**  (**UEM\_SYNC1\_ON\_DATA**)

*По умолчанию: Синхроимпульс слова данных.*

### Значения параметра UEM\_SYNC1\_ERR

* #define **UEM\_SYNC1\_ON\_ERROR**  1

*При наличии ошибки в слове.*

* #define **UEM\_SYNC1\_ON\_NO\_ERROR**  0

*При отсутствии ошибки в слове.*

* #define **UEM\_SYNC1\_ERR\_DEFAULT**  (**UEM\_SYNC1\_ON\_NO\_ERROR**)

*По умолчанию: При отсутствии ошибки в слове.*

### Значения параметра UEM\_SYNC1\_GAPB

* #define **UEM\_SYNC1\_ON\_GAPB**  1

*При наличии паузы перед словом.*

* #define **UEM\_SYNC1\_ON\_NO\_GAPB**  0

*При отсутствии паузы перед словом.*

* #define **UEM\_SYNC1\_GAPB\_DEFAULT**  (**UEM\_SYNC1\_ON\_NO\_GAPB**)

*По умолчанию: При отсутствии паузы перед словом.*

### Значения параметра UEM\_SYNC1\_CH

* #define **UEM\_SYNC1\_CH\_A**  0

*Слово передается по шине А.*

* #define **UEM\_SYNC1\_CH\_B**  2

*Слово передается по шине Б.*

* #define **UEM\_SYNC1\_ACH**  1

*Слово передается по любой шине.*

* #define **UEM\_SYNC1\_CH\_DEFAULT**  (**UEM\_SYNC1\_CH\_A**)

*Значение* ***UEM\_SYNC1\_CH*** *по умолчанию: по шине А.*

### Значения параметров UEM\_IST1, UEM\_IST2

* #define **UEM\_IST\_MIN**  1

*Минимальное значение.*

* #define **UEM\_IST\_MAX**  65536

*Максимальное значение.*

* #define **UEM\_IST\_DEFAULT**  (**UEM\_IST\_MAX**)

*Значение по умолчанию.*

### Значения параметров UEM\_BM\_WORD\_PATTERN, UEM\_BM\_WORD\_MASK

* #define **UEM\_BM\_WORD\_MIN**  0x0000

*Минимальное значение.*

* #define **UEM\_BM\_WORD\_MAX**  0xFFFF

*Максимальное значение.*

* #define **UEM\_BM\_WORD\_DEFAULT**  (**UEM\_BM\_WORD\_MIN**)

*Значение по умолчанию.*

### Подробное описание

Определения констант - значений параметров конфигурации.

Константы - значения параметров конфигурации - определены для задания значений параметров конфигурации в более понятной форме, а также показывают значения параметров по умолчанию. Эти константы предназначены для использования в качестве значений аргумента **value** в функции **uem\_param\_set()**. Их также можно использовать для сравнения со значениями, возвращаемыми в аргументе **value** в функции **uem\_param\_get()**. Использование этих констант не обязательно.

### Макросы

#### #define UEM\_TMTA\_DISABLED  1

Передатчик шины А отключен.

Используется с **UEM\_TMTA\_DIS**.

#### #define UEM\_TMTA\_ENABLED  0

Передатчик шины А включен.

Используется с **UEM\_TMTA\_DIS**.

#### #define UEM\_TMTA\_DEFAULT  (UEM\_TMTA\_ENABLED)

По умолчанию: Передатчик шины А включен.

Используется с **UEM\_TMTA\_DIS**.

#### #define UEM\_TMTB\_DISABLED  1

Передатчик шины Б отключен.

Используется с **UEM\_TMTB\_DIS**.

#### #define UEM\_TMTB\_ENABLED  0

Передатчик шины Б включен.

Используется с **UEM\_TMTB\_DIS**.

#### #define UEM\_TMTB\_DEFAULT  (UEM\_TMTB\_ENABLED)

По умолчанию: Передатчик шины Б включен.

Используется с **UEM\_TMTB\_DIS**.

#### #define UEM\_RCVA\_DISABLED  1

Приемник шины А отключен.

Используется с **UEM\_RCVA\_DIS**.

#### #define UEM\_RCVA\_ENABLED  0

Приемник шины А включен.

Используется с **UEM\_RCVA\_DIS**.

#### #define UEM\_RCVA\_DEFAULT  (UEM\_RCVA\_ENABLED)

По умолчанию: Приемник шины А включен.

Используется с **UEM\_RCVA\_DIS**.

#### #define UEM\_RCVB\_DISABLED  1

Приемник шины Б отключен.

Используется с **UEM\_RCVB\_DIS**.

#### #define UEM\_RCVB\_ENABLED  0

Приемник шины Б включен.

Используется с **UEM\_RCVB\_DIS**.

#### #define UEM\_RCVB\_DEFAULT  (UEM\_RCVB\_ENABLED)

По умолчанию: Приемник шины Б включен.

Используется с **UEM\_RCVB\_DIS**.

#### #define UEM\_SYNC\_IN\_1\_DISABLED  0

Обработка входного сигнала синхронизации 1 запрещена.

Используется с **UEM\_SYNC\_IN\_1\_ENA**.

#### #define UEM\_SYNC\_IN\_1\_ENABLED  1

Обработка входного сигнала синхронизации 1 разрешена.

Используется с **UEM\_SYNC\_IN\_1\_ENA**.

#### #define UEM\_SYNC\_IN\_1\_DEFAULT  (UEM\_SYNC\_IN\_1\_DISABLED)

По умолчанию: Обработка входного сигнала синхронизации 1 запрещена.

Используется с **UEM\_SYNC\_IN\_1\_ENA**.

#### #define UEM\_SYNC\_IN\_2\_DISABLED  0

Обработка входного сигнала синхронизации 2 запрещена.

Используется с **UEM\_SYNC\_IN\_2\_ENA**.

#### #define UEM\_SYNC\_IN\_2\_ENABLED  1

Обработка входного сигнала синхронизации 2 разрешена.

Используется с **UEM\_SYNC\_IN\_2\_ENA**.

#### #define UEM\_SYNC\_IN\_2\_DEFAULT  (UEM\_SYNC\_IN\_2\_DISABLED)

По умолчанию: Обработка входного сигнала синхронизации 2 запрещена.

Используется с **UEM\_SYNC\_IN\_2\_ENA**.

#### #define UEM\_SYNC\_OUT\_1\_DISABLED  0

Формирование выходного сигнала синхронизации 1 запрещено.

Используется с **UEM\_SYNC\_OUT\_1\_ENA**.

#### #define UEM\_SYNC\_OUT\_1\_ENABLED  1

Формирование выходного сигнала синхронизации 1 разрешено.

Используется с **UEM\_SYNC\_OUT\_1\_ENA**.

#### #define UEM\_SYNC\_OUT\_1\_DEFAULT  (UEM\_SYNC\_OUT\_1\_DISABLED)

По умолчанию: Формирование выходного сигнала синхронизации 1 запрещено.

Используется с **UEM\_SYNC\_OUT\_1\_ENA**.

#### #define UEM\_SYNC\_OUT\_2\_DISABLED  0

Формирование выходного сигнала синхронизации 2 запрещено.

Используется с **UEM\_SYNC\_OUT\_2\_ENA**.

#### #define UEM\_SYNC\_OUT\_2\_ENABLED  1

Формирование выходного сигнала синхронизации 2 разрешено.

Используется с **UEM\_SYNC\_OUT\_2\_ENA**.

#### #define UEM\_SYNC\_OUT\_2\_DEFAULT  (UEM\_SYNC\_OUT\_2\_DISABLED)

По умолчанию: Формирование выходного сигнала синхронизации 2 запрещено.

Используется с **UEM\_SYNC\_OUT\_2\_ENA**.

#### #define UEM\_BRCST\_DISABLED  1

Групповые сообщения запрещены.

Используется с **UEM\_BRCST\_DIS**.

#### #define UEM\_BRCST\_ENABLED  0

Групповые сообщения разрешены.

Используется с **UEM\_BRCST\_DIS**.

#### #define UEM\_BRCST\_DEFAULT  (UEM\_BRCST\_ENABLED)

По умолчанию: Групповые сообщения разрешены.

Используется с **UEM\_BRCST\_DIS**.

#### #define UEM\_SYNC\_IN\_1\_INTGEN\_DISABLED  0

Внутренняя генерация запрещена.

Используется с **UEM\_SYNC\_IN\_1\_INTGEN**.

#### #define UEM\_SYNC\_IN\_1\_INTGEN\_ENABLED  1

Внутренняя генерация разрешена.

Используется с **UEM\_SYNC\_IN\_1\_INTGEN**.

#### #define UEM\_SYNC\_IN\_1\_INTGEN\_DEFAULT  (UEM\_SYNC\_IN\_1\_INTGEN\_DISABLED)

По умолчанию: запрещена.

Используется с **UEM\_SYNC\_IN\_1\_INTGEN**.

#### #define UEM\_SYNC\_IN\_2\_INTGEN\_DISABLED  0

Внутренняя генерация запрещена.

Используется с **UEM\_SYNC\_IN\_2\_INTGEN**.

#### #define UEM\_SYNC\_IN\_2\_INTGEN\_ENABLED  1

Внутренняя генерация разрешена.

Используется с **UEM\_SYNC\_IN\_2\_INTGEN**.

#### #define UEM\_SYNC\_IN\_2\_INTGEN\_DEFAULT  (UEM\_SYNC\_IN\_2\_INTGEN\_DISABLED)

По умолчанию: запрещена.

Используется с **UEM\_SYNC\_IN\_2\_INTGEN**.

#### #define UEM\_ERR\_INJ\_DISABLED  1

Внесение ошибок кодирования запрещено.

Используется с **UEM\_ERR\_INJ\_DIS**.

#### #define UEM\_ERR\_INJ\_ENABLED  0

Внесение ошибок кодирования разрешено.

Используется с **UEM\_ERR\_INJ\_DIS**.

#### #define UEM\_ERR\_INJ\_DEFAULT  (UEM\_ERR\_INJ\_ENABLED)

По умолчанию: Внесение ошибок кодирования разрешено.

Используется с **UEM\_ERR\_INJ\_DIS**.

#### #define UEM\_SET  1

Установка параметра в 1.

Используется с **UEM\_TMT\_RES**, **UEM\_SYNC\_IN\_1\_SET**, **UEM\_SYNC\_IN\_2\_SET**, **UEM\_SYNC\_OUT\_1\_SET**, **UEM\_SYNC\_OUT\_2\_SET**.

#### #define UEM\_DB\_INACTIVE  0

Нет активности на шине данных.

Используется с **UEM\_DB\_ACT**.

#### #define UEM\_DB\_ACTIVE  1

Обнаружена активность на шине данных.

Используется с **UEM\_DB\_ACT**.

#### #define UEM\_RFT\_MIN  0

Минимальное значение.

Используется с **UEM\_TXA\_RFT**, **UEM\_TXB\_RFT**.

#### #define UEM\_RFT\_MAX  31

Максимальное значение.

Используется с **UEM\_TXA\_RFT**, **UEM\_TXB\_RFT**.

#### #define UEM\_RFT\_SINE  32

Специальное значение. Включает формирование синусоидального сигнала.

Используется с **UEM\_TXA\_RFT**, **UEM\_TXB\_RFT**.

#### #define UEM\_RFT\_DEFAULT  18

Значение по умолчанию (18).

Используется с **UEM\_TXA\_RFT**, **UEM\_TXB\_RFT**.

#### #define UEM\_VPP\_MIN  0x00E

Минимальное значение.

Используется с **UEM\_TXA\_VPP**, **UEM\_TXB\_VPP**.

#### #define UEM\_VPP\_MAX  0x01FF

Максимальное значение.

Используется с **UEM\_TXA\_VPP**, **UEM\_TXB\_VPP**.

#### #define UEM\_VPP\_DEFAULT  0x0136

Значение по умолчанию (0x0136).

Используется с **UEM\_TXA\_VPP**, **UEM\_TXB\_VPP**.

#### #define UEM\_MC\_DISABLED  1

Команды управления запрещены.

Используется с **UEM\_MC\_DIS**.

#### #define UEM\_MC\_ENABLED  0

Команды управления разрешены.

Используется с **UEM\_MC\_DIS**.

#### #define UEM\_MC\_DEFAULT  (UEM\_MC\_ENABLED)

По умолчанию: Команды управления разрешены.

Используется с **UEM\_MC\_DIS**.

#### #define UEM\_BTMT\_DISABLED  1

Блокирование и разблокирование передатчиков ОУ запрещено.

Используется с **UEM\_BTMT\_DIS**.

#### #define UEM\_BTMT\_ENABLED  0

Блокирование и разблокирование передатчиков ОУ разрешено.

Используется с **UEM\_BTMT\_DIS**.

#### #define UEM\_BTMT\_DEFAULT  (UEM\_BTMT\_ENABLED)

По умолчанию: Блокирование и разблокирование передатчиков ОУ разрешено.

Используется с **UEM\_BTMT\_DIS**.

#### #define UEM\_BRTF\_DISABLED  1

Блокирование и разблокирование признака неисправности ОУ запрещено.

Используется с **UEM\_BRTF\_DIS**.

#### #define UEM\_BRTF\_ENABLED  0

Блокирование и разблокирование признака неисправности ОУ разрешено.

Используется с **UEM\_BRTF\_DIS**.

#### #define UEM\_BRTF\_DEFAULT  (UEM\_BRTF\_ENABLED)

По умолчанию: Блокирование и разблокирование признака неисправности ОУ разрешено.

Используется с **UEM\_BRTF\_DIS**.

#### #define UEM\_SYNC2\_VRTA\_MIN  0

Минимальное значение.

Используется с **UEM\_SYNC2\_VRTA**.

#### #define UEM\_SYNC2\_VRTA\_MAX  31

Максимальное значение.

Используется с **UEM\_SYNC2\_VRTA**.

#### #define UEM\_SYNC2\_VRTA\_DEFAULT  0

Значение по умолчанию.

Используется с **UEM\_SYNC2\_VRTA**.

#### #define UEM\_SYNC1\_ON\_COMMAND  1

Синхроимпульс командного/ответного слова.

Используется с **UEM\_SYNC1\_C\_D\_**.

#### #define UEM\_SYNC1\_ON\_DATA  0

Синхроимпульс слова данных.

Используется с **UEM\_SYNC1\_C\_D\_**.

#### #define UEM\_SYNC1\_C\_D\_DEFAULT  (UEM\_SYNC1\_ON\_DATA)

По умолчанию: Синхроимпульс слова данных.

Используется с **UEM\_SYNC1\_C\_D\_**.

#### #define UEM\_SYNC1\_ON\_ERROR  1

При наличии ошибки в слове.

Используется с **UEM\_SYNC1\_ERR**.

#### #define UEM\_SYNC1\_ON\_NO\_ERROR  0

При отсутствии ошибки в слове.

Используется с **UEM\_SYNC1\_ERR**.

#### #define UEM\_SYNC1\_ERR\_DEFAULT  (UEM\_SYNC1\_ON\_NO\_ERROR)

По умолчанию: При отсутствии ошибки в слове.

Используется с **UEM\_SYNC1\_ERR**.

#### #define UEM\_SYNC1\_ON\_GAPB  1

При наличии паузы перед словом.

Используется с **UEM\_SYNC1\_GAPB**.

#### #define UEM\_SYNC1\_ON\_NO\_GAPB  0

При отсутствии паузы перед словом.

Используется с **UEM\_SYNC1\_GAPB**.

#### #define UEM\_SYNC1\_GAPB\_DEFAULT  (UEM\_SYNC1\_ON\_NO\_GAPB)

По умолчанию: При отсутствии паузы перед словом.

Используется с **UEM\_SYNC1\_GAPB**.

#### #define UEM\_SYNC1\_CH\_A  0

Слово передается по шине А.

Используется с **UEM\_SYNC1\_CH**.

#### #define UEM\_SYNC1\_CH\_B  2

Слово передается по шине Б.

Используется с **UEM\_SYNC1\_CH**.

#### #define UEM\_SYNC1\_ACH  1

Слово передается по любой шине.

Используется с **UEM\_SYNC1\_CH**.

#### #define UEM\_SYNC1\_CH\_DEFAULT  (UEM\_SYNC1\_CH\_A)

Значение **UEM\_SYNC1\_CH** по умолчанию: по шине А.

Используется с **UEM\_SYNC1\_CH**.

#### #define UEM\_IST\_MIN  1

Минимальное значение.

Используется с **UEM\_IST1**, **UEM\_IST2**.

#### #define UEM\_IST\_MAX  65536

Максимальное значение.

Используется с **UEM\_IST1**, **UEM\_IST2**.

#### #define UEM\_IST\_DEFAULT  (UEM\_IST\_MAX)

Значение по умолчанию.

Используется с **UEM\_IST1**, **UEM\_IST2**.

#### #define UEM\_BM\_WORD\_MIN  0x0000

Минимальное значение.

Используется с **UEM\_BM\_WORD\_PATTERN**, **UEM\_BM\_WORD\_MASK**.

#### #define UEM\_BM\_WORD\_MAX  0xFFFF

Максимальное значение.

Используется с **UEM\_BM\_WORD\_PATTERN**, **UEM\_BM\_WORD\_MASK**.

#### #define UEM\_BM\_WORD\_DEFAULT  (UEM\_BM\_WORD\_MIN)

Значение по умолчанию.

Используется с **UEM\_BM\_WORD\_PATTERN**, **UEM\_BM\_WORD\_MASK**.

## Параметры интервалов времени

Описания параметров временных интервалов и функций для работы с ними.

### Макросы

* #define **UEM\_MIN\_T1\_DEFAULT**  (4\*4)

*Минимальная пауза между командным и ответным сегментами (min t1), значение по умолчанию.*

* #define **UEM\_MIN\_T2\_DEFAULT**  (4\*4)

*Минимальная пауза между сообщениями (min t2), значение по умолчанию.*

* #define **UEM\_RTMO\_DEFAULT**  (14\*4)

*Таймаут ответа ОУ, значение по умолчанию.*

### Перечисления

* enum **UEM\_TIME\_PARAM** { **UEM\_MIN\_T1**, **UEM\_MIN\_T2**, **UEM\_RTMO** }

*Идентификатор (селектор) параметра интервала времени*

### Функции

* ViStatus **uem\_timing\_set** (**UEM\_DEVHANDLE** anydev, **UEM\_TIME\_PARAM** param, **UEM\_DWORD** value)

*Установка параметра интервала времени.*

* ViStatus **uem\_timing\_get** (**UEM\_DEVHANDLE** anydev, **UEM\_TIME\_PARAM** param, **UEM\_DWORD** \*value)

*Считывание параметра интервала времени.*

### Подробное описание

Описания параметров временных интервалов и функций для работы с ними.

Параметры интервалов времени - это нормативные значения интервалов времени между сообщениями или частями сообщений. Имеется три таких параметра:

* минимальная пауза между командным и ответным сегментами min t1,
* минимальная пауза между сообщениями min t2,
* максимальное время ожидания (таймаут) ответа ОУ.

Смысл и значения этих параметров определяются в [**1**].

В отличие от конфигурационных параметров (см. **Параметры конфигурации УЭМ**), параметры интервалов времени:

* являются параметрами программного обеспечения, а не аппаратуры,
* могут быть заданы как для УЭМ в целом, так и раздельно для виртуальных устройств (КШ/ОУ/МШ) в составе УЭМ.

В виртуальном КШ параметры интервалов времени используются для расчета стандартных пауз между сообщениями. В виртуальном ОУ параметр min t1 используется как значение стандартной паузы перед передачей ответного сегмента. В виртуальном МШ все три параметра используются в алгоритме разбора трассы, для разделения потока слов на сообщения и детектирования ошибок нарушения минимальных и максимальных пауз.

Значение задается в единицах, равных 0,25 мкс.

По умолчанию параметры интервалов времени заданы в соответствии с ГОСТ [**1**].

Для специальных целей параметры интервалов времени могут быть установлены в нестандартные значения.

### Макросы

#### #define UEM\_MIN\_T1\_DEFAULT  (4\*4)

Минимальная пауза между командным и ответным сегментами (min t1), значение по умолчанию.

#### #define UEM\_MIN\_T2\_DEFAULT  (4\*4)

Минимальная пауза между сообщениями (min t2), значение по умолчанию.

#### #define UEM\_RTMO\_DEFAULT  (14\*4)

Таймаут ответа ОУ, значение по умолчанию.

### Перечисления

#### enum UEM\_TIME\_PARAM

Идентификатор (селектор) параметра интервала времени.

**Элементы перечислений:**

***UEM\_MIN\_T1*** Минимальная пауза между командным и ответным сегментами (min t1).

***UEM\_MIN\_T2*** Минимальная пауза между сообщениями (min t2).

***UEM\_RTMO*** Таймаут ответа ОУ.

### Функции

#### ViStatus uem\_timing\_set (UEM\_DEVHANDLE *anydev*, UEM\_TIME\_PARAM *param*, UEM\_DWORD *value*)

Установка параметра интервала времени.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *anydev* | Дескриптор устройства или виртуального устройства. |
| in | *param* | Идентификатор параметра (элемент перечисления **UEM\_TIME\_PARAM**). |
| in | *value* | Значение параметра. В единицах по 0,25 мкс. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

При задании в аргументе **anydev** дескриптора виртуального устройства указанный параметр устанавливается для указанного виртуального устройства.

При задании в аргументе **anydev** дескриптора УЭМ указанный параметр устанавливается для всех виртуальных устройств этого УЭМ.

#### ViStatus uem\_timing\_get (UEM\_DEVHANDLE *anydev*, UEM\_TIME\_PARAM *param*, UEM\_DWORD \* *value*)

Считывание параметра интервала времени.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *anydev* | Дескриптор устройства или виртуального устройства. |
| in | *param* | Идентификатор параметра (элемент перечисления **UEM\_TIME\_PARAM**). |
| out | *value* | Значение параметра. В единицах по 0,25 мкс. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

## Встроенный счетчик времени

Функции для работы со встроенным счетчиком времени и метками времени.

### Структуры данных

* union **UEM\_TIME\_TAG**

Формат метки времени

### *Формат метки времени*

### Определения типов

* typedef ViUInt64 **UEM\_TIME\_TAG\_LIN**

*Метка времени в линейном формате.*

### Функции

* ViStatus **uem\_time\_tag\_get** (**UEM\_DEVHANDLE** uem, **UEM\_TIME\_TAG** \*time\_tag)

*Считывание встроенного счетчика времени.*

* ViStatus **uem\_time\_tag\_set** (**UEM\_DEVHANDLE** uem, **UEM\_TIME\_TAG** \*time\_tag)

*Установка значения встроенного счетчика времени.*

* ViStatus **uem\_time\_tag\_reset** (**UEM\_DEVHANDLE** uem)

*Сброс встроенного счетчика времени.*

* **UEM\_TIME\_TAG\_LIN** **uem\_time\_tag\_to\_linear** (**UEM\_TIME\_TAG** \*time\_tag)

*Перевод метки времени в линейный формат.*

* void **uem\_time\_tag\_to\_struct** (**UEM\_TIME\_TAG** \*time\_tag, **UEM\_TIME\_TAG\_LIN** linear)

*Перевод метки времени из линейного в структурированный формат.*

### Подробное описание

Функции для работы со встроенным счетчиком времени и метками времени.

В УЭМ имеется встроенный счетчик времени с разрешением 0,25 мкс на единицу младшего разряда и диапазоном 366 суток.

В этом разделе предоставляются функции считывания и установки значения счетчика времени. Эти значения будем называть метками времени. Это структурированные значения, описываемые типом данных **UEM\_TIME\_TAG**. Метки времени в таком же формате используются в функциях монитора шины.

Рекомендуется в начале работы приложения с УЭМ установить счетчик времени в некоторое определенной значение. Это лучше всего сделать одним из следующих способов:

* установить счетчик времени в значение, соответствующее текущей дате и времени (при помощи **uem\_time\_tag\_set()**), в этом случае счетчик времени показывает текущее время,
* установить счетчик времени в значение 0 (при помощи **uem\_time\_tag\_reset()**), в этом случае счетчик времени показывает интервал времени, прошедший с момента запуска приложения.

Для вычислений с метками времени иногда удобнее не структурированный, а линейный формат, который представляет собой простое количество единиц времени по 0,25 мкс, прошедших с момента заданного "начала времени". Для перевода метки времени в линейный формат и обратно служат функции **uem\_time\_tag\_to\_linear()** и **uem\_time\_tag\_to\_struct()**.

### Типы

#### typedef ViUInt64 UEM\_TIME\_TAG\_LIN

Метка времени в линейном формате.

В единицах по 0,25 мкс.

### Функции

#### ViStatus uem\_time\_tag\_get (UEM\_DEVHANDLE *uem*, UEM\_TIME\_TAG \* *time\_tag*)

Считывание встроенного счетчика времени.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *uem* | - Дескриптор УЭМ.  Допускается указывать вместо дескриптора УЭМ дескриптор любого виртуального устройства в составе этого УЭМ. Функция выполняет такой вызов, как если бы был указан дескриптор УЭМ. |
| out | *time\_tag* | - В этом аргументе передается адрес структуры, в которую будет записано текущее значение счетчика. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_time\_tag\_set (UEM\_DEVHANDLE *uem*, UEM\_TIME\_TAG \* *time\_tag*)

Установка значения встроенного счетчика времени.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *uem* | - Дескриптор УЭМ.  Допускается указывать вместо дескриптора УЭМ дескриптор любого виртуального устройства в составе этого УЭМ. Функция выполняет такой вызов, как если бы был указан дескриптор УЭМ. |
| in | *time\_tag* | - В этом аргументе передается адрес структуры, из которой берется новое значение счетчика времени. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_time\_tag\_reset (UEM\_DEVHANDLE *uem*)

Сброс встроенного счетчика времени.

Встроенный счетчик устанавливается в значение 0.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *uem* | - Дескриптор УЭМ.  Допускается указывать вместо дескриптора УЭМ дескриптор любого виртуального устройства в составе этого УЭМ. Функция выполняет такой вызов, как если бы был указан дескриптор УЭМ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### UEM\_TIME\_TAG\_LIN uem\_time\_tag\_to\_linear (UEM\_TIME\_TAG \* *time\_tag*)

Перевод метки времени в линейный формат.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *time\_tag* | - В этом аргументе передается адрес структуры, содержащей значение метки времени. |

##### Возвращает:

Значение метки времени в линейном формате.

#### void uem\_time\_tag\_to\_struct (UEM\_TIME\_TAG \* *time\_tag*, UEM\_TIME\_TAG\_LIN *linear*)

Перевод метки времени из линейного в структурированный формат.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| out | *time\_tag* | - В этом аргументе передается адрес структуры, в которую будет записано значение метки времени. |
| in | *linear* | - Значение метки времени в линейном формате. |

##### Возвращает:

Функция не возвращает значения.

## Определения типов данных для КШ, ОУ, МШ

Определения типов данных для КШ, ОУ, МШ.

### Структуры данных

* struct **UEM\_DATA**

*Блок слов данных.*

* struct **UEM\_CMD\_SEG**

*Образ командного сегмента.*

* struct **UEM\_RESP\_SEG**

*Ответный сегмент.*

* struct **UEM\_RAW\_BM\_MESSAGE**

*Принятое сообщение в аппаратном формате.*

* struct **UEM\_SEGMENT\_DESCR**

*Описатель сегмента в мониторе шины.*

* struct **UEM\_BM\_MESSAGE**

### *Разобранное сообщение МШ.*

### Определения типов

* typedef ViUInt32 **UEM\_ERROR\_FLAGS**

***Признаки ошибок распознавания сообщения*** *в мониторе шины.*

### Перечисления

* enum **UEM\_FORMAT** { **UEM\_UNF**, **UEM\_F1**, **UEM\_F2**, **UEM\_F3**, **UEM\_F4**, **UEM\_F5**, **UEM\_F6**, **UEM\_F7**, **UEM\_F8**, **UEM\_F9**, **UEM\_F10** }

*Форматы сообщений (номера по ГОСТ [****1****]).*

* enum **UEM\_CHANNEL** { **UEM\_CH\_A**, **UEM\_CH\_B** }

*Селектор шины (А/Б).*

* enum **UEM\_SYNC** { **UEM\_SYNC\_D**, **UEM\_SYNC\_C** }

*Селектор синхроимпульса.*

### Более мнемоничные обозначения форматов сообщений

* #define **UEM\_BCRT**  **UEM\_F1**

*КШОУ.*

* #define **UEM\_RTBC**  **UEM\_F2**

*ОУКШ.*

* #define **UEM\_RTRT**  **UEM\_F3**

*ОУОУ.*

* #define **UEM\_MC**  **UEM\_F4**

*Команда управления.*

* #define **UEM\_MCRTBC**  **UEM\_F5**

*Команда управления со словом данных, передаваемым от ОУ к КШ.*

* #define **UEM\_MCBCRT**  **UEM\_F6**

*Команда управления со словом данных, передаваемым от КШ к ОУ.*

* #define **UEM\_BCRTb**  **UEM\_F7**

*КШОУ ГРУППОВОЕ.*

* #define **UEM\_RTRTb**  **UEM\_F8**

*ОУОУ ГРУППОВОЕ.*

* #define **UEM\_MCb**  **UEM\_F9**

*Команда управления групповая.*

* #define **UEM\_MCBCRTb**  **UEM\_F10**

*Команда управления групповая со словом данных, передаваемым от КШ к ОУ.*

### Число слов данных

* #define **UEM\_NDATA\_MIN**  0

*Число слов данных - минимальное значение.*

* #define **UEM\_NDATA\_MAX**  62

*Число слова данных - максимальное значение.*

* #define **UEM\_NDATA\_BY\_CW**  63

*Число слов данных определяется командным словом.*

### Признаки ошибок распознавания сообщения

Комбинации перечисленных констант (по |) определяют содержание элементов типа **UEM\_ERROR\_FLAGS**.

* #define **UEM\_ERRF\_ERROR**  1

*Наличие любой ошибки (суммарный флаг).*

* #define **UEM\_ERRF\_ENCODING**  (1<<1)

*Наличие ошибки кодирования слов (суммарный флаг).*

* #define **UEM\_ERRF\_FORMAT**  (1<<2)

*Наличие нарушения формата, ошибки состава сообщения (суммарный флаг).*

* #define **UEM\_ERRF\_MINGAP**  (1<<3)

*Временной интервал меньше допустимого.*

* #define **UEM\_ERRF\_NO\_RESPONSE**  (1<<4)

*Отсутствие ответа.*

* #define **UEM\_ERRF\_SYNC\_TYPE**  (1<<5)

*Неверный тип синхроимпульса.*

* #define **UEM\_ERRF\_MISSING\_CWSW**  (1<<6)

*Отсутствует командное или ответное слово.*

* #define **UEM\_ERRF\_EXTRA\_CWSW**  (1<<7)

*Лишнее командное или ответное слово.*

* #define **UEM\_ERRF\_MISSING\_DW**  (1<<8)

*Недостаточно слов данных.*

* #define **UEM\_ERRF\_EXTRA\_DW**  (1<<9)

*Лишние слова данных.*

* #define **UEM\_ERRF\_INCORRECT\_RTN**  (1<<10)

*Некорректный адрес ОУ.*

* #define **UEM\_ERRF\_RTRT\_FORMAT**  (1<<11)

*Ошибка формата ОУОУ (одинаковые адреса ОУ, несовпадение числа СД).*

* #define **UEM\_ERRF\_INC\_MODE\_CODE**  (1<<12)

*Некорректная команда управления.*

* #define **UEM\_ERRF\_FORMAT\_MC**  (1<<13)

*Ошибка формата команды управления.*

* #define **UEM\_ERRF\_GAPN**  (1<<20)

*Недостоверная информация (сигнал) во время паузы перед словом.*

* #define **UEM\_ERRF\_PARITY**  (1<<21)

*Ошибка четности.*

* #define **UEM\_ERRF\_LESS\_BITS**  (1<<22)

*Укороченное слово.*

* #define **UEM\_ERRF\_MORE\_BITS**  (1<<23)

*Удлиненное слово.*

* #define **UEM\_ERRF\_ENC**  (1<<24)

*Ошибка бифазного кодирования.*

* #define **UEM\_ERRF\_DT**  (1<<30)

*Несоблюдение минимальной паузы перед словом, по данным аппаратного декодера.*

* #define **UEM\_ERRF\_ENCODING2**  (1<<31)

*Наличие любой ошибки кодирования слов (суммарный флаг, по данным аппаратного декодера).*

### Подробное описание

Определения типов данных для КШ, ОУ, МШ.

Данный раздел содержит определения структур и типов данных, ориентированных на протокол МКПД [**1**] и использующихся в функциях виртуальных КШ, ОУ, МШ.

### Макросы

#### #define UEM\_BCRT  UEM\_F1

КШОУ.

#### #define UEM\_RTBC  UEM\_F2

ОУКШ.

#### #define UEM\_RTRT  UEM\_F3

ОУОУ.

#### #define UEM\_MC  UEM\_F4

Команда управления.

#### #define UEM\_MCRTBC  UEM\_F5

Команда управления со словом данных, передаваемым от ОУ к КШ.

#### #define UEM\_MCBCRT  UEM\_F6

Команда управления со словом данных, передаваемым от КШ к ОУ.

#### #define UEM\_BCRTb  UEM\_F7

КШОУ ГРУППОВОЕ.

#### #define UEM\_RTRTb  UEM\_F8

ОУОУ ГРУППОВОЕ.

#### #define UEM\_MCb  UEM\_F9

Команда управления групповая.

#### #define UEM\_MCBCRTb  UEM\_F10

Команда управления групповая со словом данных, передаваемым от КШ к ОУ.

#### #define UEM\_NDATA\_MIN  0

Число слов данных - минимальное значение.

#### #define UEM\_NDATA\_MAX  62

Число слова данных - максимальное значение.

**Примечание:** Аппаратно УЭМ может формировать командные и ответные сегменты, содержащие до 62 слов данных.

#### #define UEM\_NDATA\_BY\_CW  63

Число слов данных определяется командным словом.

Используется в **uem\_response\_create()**.

#### #define UEM\_ERRF\_ERROR  1

Наличие любой ошибки (суммарный флаг).

#### #define UEM\_ERRF\_ENCODING  (1<<1)

Наличие ошибки кодирования слов (суммарный флаг).

#### #define UEM\_ERRF\_FORMAT  (1<<2)

Наличие нарушения формата, ошибки состава сообщения (суммарный флаг).

#### #define UEM\_ERRF\_MINGAP  (1<<3)

Временной интервал меньше допустимого.

#### #define UEM\_ERRF\_NO\_RESPONSE  (1<<4)

Отсутствие ответа.

#### #define UEM\_ERRF\_SYNC\_TYPE  (1<<5)

Неверный тип синхроимпульса.

#### #define UEM\_ERRF\_MISSING\_CWSW  (1<<6)

Отсутствует командное или ответное слово.

#### #define UEM\_ERRF\_EXTRA\_CWSW  (1<<7)

Лишнее командное или ответное слово.

#### #define UEM\_ERRF\_MISSING\_DW  (1<<8)

Недостаточно слов данных.

#### #define UEM\_ERRF\_EXTRA\_DW  (1<<9)

Лишние слова данных.

#### #define UEM\_ERRF\_INCORRECT\_RTN  (1<<10)

Некорректный адрес ОУ.

#### #define UEM\_ERRF\_RTRT\_FORMAT  (1<<11)

Ошибка формата ОУОУ (одинаковые адреса ОУ, несовпадение числа СД).

#### #define UEM\_ERRF\_INC\_MODE\_CODE  (1<<12)

Некорректная команда управления.

#### #define UEM\_ERRF\_FORMAT\_MC  (1<<13)

Ошибка формата команды управления.

#### #define UEM\_ERRF\_GAPN  (1<<20)

Недостоверная информация (сигнал) во время паузы перед словом.

#### #define UEM\_ERRF\_PARITY  (1<<21)

Ошибка четности.

#### #define UEM\_ERRF\_LESS\_BITS  (1<<22)

Укороченное слово.

#### #define UEM\_ERRF\_MORE\_BITS  (1<<23)

Удлиненное слово.

#### #define UEM\_ERRF\_ENC  (1<<24)

Ошибка бифазного кодирования.

#### #define UEM\_ERRF\_DT  (1<<30)

Несоблюдение минимальной паузы перед словом, по данным аппаратного декодера.

#### #define UEM\_ERRF\_ENCODING2  (1<<31)

Наличие любой ошибки кодирования слов (суммарный флаг, по данным аппаратного декодера).

### Типы

#### typedef ViUInt32 UEM\_ERROR\_FLAGS

**Признаки ошибок распознавания сообщения** в мониторе шины.

### Перечисления

#### enum UEM\_FORMAT

Форматы сообщений (номера по ГОСТ [**1**]).

**Элементы перечислений:**

***UEM\_UNF*** Сообщение с нарушенным форматом.

***UEM\_F1*** Формат 1.

***UEM\_F2*** Формат 2.

***UEM\_F3*** Формат 3.

***UEM\_F4*** Формат 4.

***UEM\_F5*** Формат 5.

***UEM\_F6*** Формат 6.

***UEM\_F7*** Формат 7.

***UEM\_F8*** Формат 8.

***UEM\_F9*** Формат 9.

***UEM\_F10*** Формат 10.

#### enum UEM\_CHANNEL

Селектор шины (А/Б).

**Элементы перечислений:**

***UEM\_CH\_A*** Шина А.

***UEM\_CH\_B*** Шина Б.

#### enum UEM\_SYNC

Селектор синхроимпульса.

**Элементы перечислений:**

***UEM\_SYNC\_D*** Синхроимпульс слова данных.

***UEM\_SYNC\_C*** Синхроимпульс командного и ответного слова.

## Функции КШ

Описания функций контроллера шины.

### Группы

* **Заполнение образа командного сегмента в ОЗУ ПЭВМ**

*Функции заполнения образа командного сегмента в ОЗУ ПЭВМ.*

* **Внесение ошибок состава сообщения**

*Порядок действий по внесению ошибок состава сообщения.*

* **Создание и настройка командных сегментов**

*Функции создания и настройки командных сегментов.*

* **Создание и настройка кадров и программы КШ**

*Функции создания и настройки кадров и программ КШ.*

* **Запуск и остановка КШ**

*Порядок запуска, остановки и контроля активности КШ.*

* **Передача сообщений**

*Описание функции передачи отдельного сообщения.*

### Подробное описание

Описания функций контроллера шины.

Для использования этих функций необходимо получить дескриптор виртуального КШ при помощи функции **uem\_bc\_init()**.

Программирование КШ заключается в создании в ОЗУ КШ и настройке специальных объектов:

* командных сегментов (см. **Командные и ответные сегменты**),
* кадров - последовательностей командных сегментов,
* программ КШ - последовательностей кадров.

Командные сегменты создаются в 2 этапа:

1. Формируется образ сегмента в ОЗУ управляющей ПЭВМ в соответствии с форматом сообщения МКПД.

2. На основе этого образа создается командный сегмент в ОЗУ КШ.

После создания в командные сегменты могут быть внесены ошибки различного рода.

Кадры и программа КШ определяют состав и условия повторения последовательностей сообщений.

В заключение конфигурирования нужно указать КШ, какая именно программа КШ является исполняемой (даже если она одна).

После этого КШ может быть запущен в работу.

## Заполнение образа командного сегмента в ОЗУ ПЭВМ

Функции заполнения образа командного сегмента в ОЗУ ПЭВМ.

### Функции

* ViStatus **uem\_bc\_cseg\_format** (**UEM\_DEVHANDLE** bc, **UEM\_CMD\_SEG** \*cseg\_data, **UEM\_CHANNEL** ch, **UEM\_FORMAT** format, **UEM\_WORD** rt, **UEM\_WORD** sa, **UEM\_WORD** ndatawords, **UEM\_WORD** \*datawords)

*Формирование образов командных сегментов для сообщений форматов 1,2,7 и неформатных сообщений.*

* ViStatus **uem\_bc\_cseg\_format\_RTRT** (**UEM\_DEVHANDLE** bc, **UEM\_CMD\_SEG** \*cseg\_data, **UEM\_CHANNEL** ch, **UEM\_FORMAT** format, **UEM\_WORD** rtrx, **UEM\_WORD** sarx, **UEM\_WORD** rttx, **UEM\_WORD** satx, **UEM\_WORD** ndatawords)

*Формирование образов командных сегментов для сообщений форматов 3,8.*

* ViStatus **uem\_bc\_cseg\_format\_MODE** (**UEM\_DEVHANDLE** bc, **UEM\_CMD\_SEG** \*cseg\_data, **UEM\_CHANNEL** ch, **UEM\_FORMAT** format, **UEM\_WORD** rt, **UEM\_WORD** mode, **UEM\_WORD** modecode, **UEM\_WORD** dataword)

*Формирование образов командных сегментов для сообщений форматов 4,5,6,9,10.*

### Подробное описание

Функции заполнения образа командного сегмента в ОЗУ ПЭВМ.

Функции **uem\_bc\_cseg\_format()**, **uem\_bc\_cseg\_format\_RTRT()**, **uem\_bc\_cseg\_format\_MODE()** заполняют командный сегмент **cseg\_data** в соответствии с заданным форматом сообщения.

* Функция **uem\_bc\_cseg\_format()** предназначена для формирования образов сегментов для сообщений форматов 1,2,7 и неформатных.
* Функция **uem\_bc\_cseg\_format\_RTRT()** предназначена для формирования образов сегментов для сообщений форматов 3 и 8.
* Функция **uem\_bc\_cseg\_format\_MODE()** предназначена для формирования образов сегментов для сообщений форматов 4,5,6,9,10.

Функции различаются набором аргументов в соответствии с форматами сообщений. На работу функций влияют значения параметров конфигурации **UEM\_MC\_DIS**, **UEM\_BRCST\_DIS**.

### Функции

#### ViStatus uem\_bc\_cseg\_format (UEM\_DEVHANDLE *bc*, UEM\_CMD\_SEG \* *cseg\_data*, UEM\_CHANNEL *ch*, UEM\_FORMAT *format*, UEM\_WORD *rt*, UEM\_WORD *sa*, UEM\_WORD *ndatawords*, UEM\_WORD \* *datawords*)

Формирование образов командных сегментов для сообщений форматов 1,2,7 и неформатных сообщений.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bc* | Дескриптор КШ. |
| out | *cseg\_data* | Образ командного сегмента в ОЗУ ПЭВМ. |
| in | *ch* | Выбор шины (А/Б). |
| in | *format* | Формат сообщения. |
| in | *rt* | Адрес ОУ. |
| in | *sa* | Подадрес. |
| in | *ndatawords* | Число слов данных. |
| in | *datawords* | Массив со словами данных. Должен быть задан в сообщениях форматов 1, 7, в неформатных сообщениях (**UEM\_UNF**), в остальных случаях игнорируется. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

В неформатных сообщениях (format == **UEM\_UNF**) аргументы **rt** и **sa** игнорируются, используются аргументы **ndatawords** и **datawords** .

#### ViStatus uem\_bc\_cseg\_format\_RTRT (UEM\_DEVHANDLE *bc*, UEM\_CMD\_SEG \* *cseg\_data*, UEM\_CHANNEL *ch*, UEM\_FORMAT *format*, UEM\_WORD *rtrx*, UEM\_WORD *sarx*, UEM\_WORD *rttx*, UEM\_WORD *satx*, UEM\_WORD *ndatawords*)

Формирование образов командных сегментов для сообщений форматов 3,8.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bc* | Дескриптор КШ. |
| out | *cseg\_data* | Образ командного сегмента в ОЗУ ПЭВМ. |
| in | *ch* | Выбор шины (А/Б). |
| in | *format* | Формат сообщения. |
| in | *rtrx* | Адрес принимающего ОУ. |
| in | *sarx* | Подадрес принимающего ОУ. |
| in | *rttx* | Адрес передающего ОУ. |
| in | *satx* | Подадрес передающего ОУ. |
| in | *ndatawords* | Число слов данных. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_bc\_cseg\_format\_MODE (UEM\_DEVHANDLE *bc*, UEM\_CMD\_SEG \* *cseg\_data*, UEM\_CHANNEL *ch*, UEM\_FORMAT *format*, UEM\_WORD *rt*, UEM\_WORD *mode*, UEM\_WORD *modecode*, UEM\_WORD *dataword*)

Формирование образов командных сегментов для сообщений форматов 4,5,6,9,10.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bc* | Дескриптор КШ. |
| out | *cseg\_data* | Образ командного сегмента в ОЗУ ПЭВМ. |
| in | *ch* | Выбор шины (А/Б). |
| in | *format* | Формат сообщения.  В дополнение к перечисленным форматам в этом аргументе допускается указывать значение **UEM\_UNF**. Это служит указанием функции подобрать нужный формат автоматически. Для указания группового формата в этом случае следует задать аргумент **rt** = 31. |
| in | *rt* | Адрес ОУ. |
| in | *mode* | Режим управления в командах управления. |
| in | *modecode* | Код команды управления. |
| in | *dataword* | Слово данных в команде управления. Игнорируется, если КУ не предполагает передачу СД в командном сегменте. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

## Внесение ошибок состава сообщения

Порядок действий по внесению ошибок состава сообщения.

Под ошибками состава сообщения понимается несоответствие числа слов данных формату сообщения и командному слову, в том числе: неверное число слов данных, наличие слов данных в сегментах, в которых их быть не должно, отсутствие слов данных в сегментах, в которых они должны быть, а также - несоответствие друг другу командных слов в сообщениях ОУОУ (форматов 3, 8), групповые адреса в командных словах, в которых они недопустимы, и некоторые другие.

Для внесения ошибок состава сообщения в командный сегмент КШ следует:

1. Заполнить образ командного сегмента корректным командным сегментом при помощи функции **uem\_bc\_cseg\_format()**, **uem\_bc\_cseg\_format\_RTRT()** или **uem\_bc\_cseg\_format\_MODE()**.

2. Изменить этот образ программно.

Альтернативно, приложение может заполнить образ командного сегмента самостоятельно с самого начала.

## Создание и настройка командных сегментов

Функции создания и настройки командных сегментов.

### Перечисления

* enum **UEM\_CSEG\_TYPE** { **UEM\_CSEG\_NORMAL**, **UEM\_CSEG\_OVERLAY**, **UEM\_CSEG\_GAP** }

*Тип командного сегмента.*

### Функции

* ViStatus **uem\_bc\_cseg\_create** (**UEM\_DEVHANDLE** bc, **UEM\_OBJHANDLE** \*cseg, **UEM\_CMD\_SEG** \*cseg\_data)

*Создание командного сегмента.*

* ViStatus **uem\_cseg\_read** (**UEM\_OBJHANDLE** cseg, **UEM\_CMD\_SEG** \*cseg\_data)

*Чтение командного сегмента.*

* ViStatus **uem\_cseg\_gap\_set** (**UEM\_OBJHANDLE** cseg, **UEM\_WORD** gap, **UEM\_WORD** gap\_flags, **UEM\_WORD** gap\_timeout)

*Программирование паузы перед сообщением.*

* ViStatus **uem\_cseg\_gap\_get** (**UEM\_OBJHANDLE** cseg, **UEM\_WORD** \*gap, **UEM\_WORD** \*gap\_flags, **UEM\_WORD** \*gap\_timeout)

*Считывание паузы перед сообщением.*

* ViStatus **uem\_cseg\_gap\_reset** (**UEM\_OBJHANDLE** cseg)

*Сброс паузы перед сообщением.*

* ViStatus **uem\_cseg\_word\_gap\_set** (**UEM\_OBJHANDLE** cseg, **UEM\_WORD** wordnumber, **UEM\_WORD** gap, **UEM\_WORD** gap\_flags, **UEM\_WORD** gap\_timeout)

*Программирование паузы между словами.*

* ViStatus **uem\_cseg\_word\_gap\_get** (**UEM\_OBJHANDLE** cseg, **UEM\_WORD** wordnumber, **UEM\_WORD** \*gap, **UEM\_WORD** \*gap\_flags, **UEM\_WORD** \*gap\_timeout)

*Считывание паузы перед словом.*

* ViStatus **uem\_cseg\_error\_set** (**UEM\_OBJHANDLE** cseg, **UEM\_WORD** wordnumber, **UEM\_ERROR\_TYPE** error\_type, ViInt32 error\_pos, ViInt32 error\_param)

*Внесение ошибок кодирования.*

* ViStatus **uem\_cseg\_error\_get** (**UEM\_OBJHANDLE** cseg, **UEM\_WORD** wordnumber, **UEM\_ERROR\_TYPE** \*error\_type, ViInt32 \*error\_pos, ViInt32 \*error\_param)

*Считывание внесенных ошибок кодирования.*

* ViStatus **uem\_cseg\_sync\_set** (**UEM\_OBJHANDLE** cseg, **UEM\_WORD** wordnumber, **UEM\_SYNC** sync)

*Установка типа синхроимпульса.*

* ViStatus **uem\_cseg\_sync\_get** (**UEM\_OBJHANDLE** cseg, **UEM\_WORD** wordnumber, **UEM\_SYNC** \*sync)

*Считывание типа синхроимпульса.*

* ViStatus **uem\_bc\_gap\_create** (**UEM\_DEVHANDLE** bc, **UEM\_OBJHANDLE** \*cseg, **UEM\_WORD** gap, **UEM\_WORD** gap\_flags, **UEM\_WORD** gap\_timeout, **UEM\_CHANNEL** ch)

*Создание паузы.*

* ViStatus **uem\_bc\_cseg\_overlay** (**UEM\_DEVHANDLE** bc, **UEM\_OBJHANDLE** \*cseg\_o, **UEM\_OBJHANDLE** cseg\_1, **UEM\_WORD** gap, **UEM\_OBJHANDLE** cseg\_2)

*Создание сообщения с наложением.*

* ViStatus **uem\_cseg\_type** (**UEM\_OBJHANDLE** cseg, **UEM\_CSEG\_TYPE** \*type)

*Запрос типа командного сегмента.*

* ViStatus **uem\_cseg\_desrtoy** (**UEM\_OBJHANDLE** cseg)

*Уничтожение командного сегмента.*

### Флаги отсчета паузы

Пауза отсчитывается от одного из определенных событий.

Флаги отсчета паузы определяют выбор события.

Пауза может отсчитываться от начала либо конца предшествующего сегмента, переданного КШ либо другим абонентом (ОУ), по той же либо по альтернативной шине. Выбор каждой из альтернатив управляется отдельным флагом. Для начала отсчета паузы также может использоваться сигнал внешней синхронизации **sync\_in\_1**. Поступление этого сигнала используется *вместо* события начала или конца сегмента, переданного КШ. Если же задан отсчет паузы от начала (или конца) сегмента, переданного другим абонентом, то учитываются оба события - начало (или конец) сегмента и поступление сигнала. При задании отсчета паузы от начала (или конца) сегмента другого абонента, или от поступления сигнала, или обоих, то есть - от внешних событий, которые теоретически могут и не наступить, используется таймаут, который задается в функции **uem\_cseg\_gap\_set()**, и по истечении которого отсчет паузы начинается без дальнейшего ожидания внешних событий.

* #define **UEM\_CSEG\_GAP\_FROM\_START**  0x20

*Отсчет паузы от начала предыдущего сегмента.*

* #define **UEM\_CSEG\_GAP\_FROM\_END**  0

*Отсчет паузы от конца предыдущего сегмента.*

* #define **UEM\_CSEG\_GAP\_ESYNC**  0x10

*Отсчет паузы после сигнала внешней синхронизации (только для КШ).*

* #define **UEM\_CSEG\_GAP\_ALT\_BUS**  0x08

*Отсчет паузы от сегмента по альтернативной шине (только для КШ).*

* #define **UEM\_CSEG\_GAP\_THIS\_BUS**  0

*Отсчет паузы от сегмента по этой же шине (только для КШ).*

* #define **UEM\_CSEG\_GAP\_ALT\_AB**  0x04

*Отсчет паузы от сегмента другого абонента (ОУ) (только для КШ).*

* #define **UEM\_CSEG\_GAP\_THIS\_AB**  0

*Отсчет паузы от собственного сегмента (только для КШ).*

* #define **UEM\_CSEG\_GAP\_DEFAULT\_FLAGS**  (**UEM\_CSEG\_GAP\_FROM\_START**)

*Стандартный набор флагов отсчета паузы.*

### Диапазоны значений параметров отсчета паузы

* #define **UEM\_CSEG\_GAP\_MIN**  0

*Минимальное значение паузы.*

* #define **UEM\_CSEG\_GAP\_MAX**  65535

*Максимальное значение паузы.*

* #define **UEM\_CSEG\_GAP\_DEFAULT\_VALUE**  0

*Значение паузы по умолчанию.*

* #define **UEM\_CSEG\_GAP\_TIMEOUT\_MIN**  0

*Минимальное значение таймаута отсчета паузы.*

* #define **UEM\_CSEG\_GAP\_TIMEOUT\_MAX**  1023

*Максимальное значение таймаута отсчета паузы.*

* #define **UEM\_CSEG\_GAP\_DEFAULT\_TIMEOUT**  0

*Таймаут отсчета паузы по умолчанию.*

### Подробное описание

Функции создания и настройки командных сегментов.

В данном разделе описаны функции создания командных сегментов и задания их дополнительных параметров, таких как паузы и ошибки кодирования.

### Макросы

#### #define UEM\_CSEG\_GAP\_FROM\_START  0x20

Отсчет паузы от начала предыдущего сегмента.

Пауза может отсчитываться от начала или окончания переданного ранее сегмента слов. Если флаг установлен (UEM\_CSEG\_GAP\_FROM\_START=1), то отсчет паузы будет вестись от начала передачи этого сегмента слов, если не установлен – от окончания переданного ранее сегмента слов. Для ОУ флаг имеет смысл только для описания СД, так как для ОС (первого слова ответного сегмента) пауза всегда отсчитывается от окончания приема командного сегмента (либо ответного сегмента передающего ОУ для принимающего ОУ в сообщениях формата ОУОУ).

#### #define UEM\_CSEG\_GAP\_FROM\_END  0

Отсчет паузы от конца предыдущего сегмента.

Данный макрос введен, чтобы явно обозначить, что флаг **UEM\_CSEG\_GAP\_FROM\_START** сброшен. Макрос не должен указываться одновременно с **UEM\_CSEG\_GAP\_FROM\_START**.

#### #define UEM\_CSEG\_GAP\_ESYNC  0x10

Отсчет паузы после сигнала внешней синхронизации (только для КШ).

Флаг **UEM\_CSEG\_GAP\_ESYNC** определяет, задается ли отсчет паузы внешним синхросигналом или моментом, связанным с передачей или приемом предыдущего сегмента, в соответствии с состояниями флагов **UEM\_CSEG\_GAP\_THIS\_BUS** / **UEM\_CSEG\_GAP\_ALT\_BUS**, **UEM\_CSEG\_GAP\_THIS\_AB** / **UEM\_CSEG\_GAP\_ALT\_AB**, **UEM\_CSEG\_GAP\_FROM\_START** / **UEM\_CSEG\_GAP\_FROM\_END**. Единичное значение (UEM\_CSEG\_GAP\_ESYNC=1) означает, что отсчет паузы начнется при поступлении внешнего синхросигнала. Нулевое значение – от момента, определяемого предыдущим сегментом. Если одновременно установлены флаги **UEM\_CSEG\_GAP\_ESYNC** и **UEM\_CSEG\_GAP\_ALT\_AB**, то для начала отсчета паузы должны произойти оба события. Синхросигнал должен поступать на вход внутренней синхронизации 1. Дополнительно в УЭМ имеются средства внутренней генерации синхросигнала 1.

См. подробнее в разделе **Управление синхронизацией**.

#### #define UEM\_CSEG\_GAP\_ALT\_BUS  0x08

Отсчет паузы от сегмента по альтернативной шине (только для КШ).

Флаг **UEM\_CSEG\_GAP\_ALT\_BUS** задает шину, событие на которой определяет момент начала отсчета паузы. Единичное значение флага (UEM\_CSEG\_GAP\_ALT\_BUS=1) означает, что отсчет паузы будет вестись от события в резервной шине относительно шины, заданной текущим значением поля **ch** в **UEM\_CMD\_SEG**. Значения полей данной структуры учитываются при создании командного сегмента функцией **uem\_bc\_cseg\_create()**. Значение поля **ch** может быть задано как вручную, так и с использованием функций **uem\_bc\_cseg\_format()**, **uem\_bc\_cseg\_format\_RTRT()**, **uem\_bc\_cseg\_format\_MODE()**. При нулевом значении данного флага отсчет паузы начинается от события в той же шине. Для ОУ данный флаг незначим.

#### #define UEM\_CSEG\_GAP\_THIS\_BUS  0

Отсчет паузы от сегмента по этой же шине (только для КШ).

Данный макрос введен, чтобы явно обозначить, что флаг **UEM\_CSEG\_GAP\_ALT\_BUS** сброшен. Макрос не должен указываться одновременно с **UEM\_CSEG\_GAP\_ALT\_BUS**.

#### #define UEM\_CSEG\_GAP\_ALT\_AB  0x04

Отсчет паузы от сегмента другого абонента (ОУ) (только для КШ).

Флаг **UEM\_CSEG\_GAP\_ALT\_AB** задает отсчет паузы от начала или конца сегмента, переданного другим абонентом (ОУ), при единичном значении. При нулевом значении отсчет паузы выполняется от начала или конца сегмента, переданного самим КШ.

#### #define UEM\_CSEG\_GAP\_THIS\_AB  0

Отсчет паузы от собственного сегмента (только для КШ).

Данный макрос введен, чтобы явно обозначить, что флаг **UEM\_CSEG\_GAP\_ALT\_AB** сброшен. Макрос не должен указываться одновременно с **UEM\_CSEG\_GAP\_ALT\_AB**.

#### #define UEM\_CSEG\_GAP\_DEFAULT\_FLAGS  (UEM\_CSEG\_GAP\_FROM\_START)

Стандартный набор флагов отсчета паузы.

#### #define UEM\_CSEG\_GAP\_MIN  0

Минимальное значение паузы.

#### #define UEM\_CSEG\_GAP\_MAX  65535

Максимальное значение паузы.

#### #define UEM\_CSEG\_GAP\_DEFAULT\_VALUE  0

Значение паузы по умолчанию.

#### #define UEM\_CSEG\_GAP\_TIMEOUT\_MIN  0

Минимальное значение таймаута отсчета паузы.

#### #define UEM\_CSEG\_GAP\_TIMEOUT\_MAX  1023

Максимальное значение таймаута отсчета паузы.

#### #define UEM\_CSEG\_GAP\_DEFAULT\_TIMEOUT  0

Таймаут отсчета паузы по умолчанию.

### Перечисления

#### enum UEM\_CSEG\_TYPE

Тип командного сегмента.

**Элементы перечислений:**

***UEM\_CSEG\_NORMAL*** Нормальный сегмент.

***UEM\_CSEG\_OVERLAY*** Сегмент с наложением.

***UEM\_CSEG\_GAP*** Пауза.

### Функции

#### ViStatus uem\_bc\_cseg\_create (UEM\_DEVHANDLE *bc*, UEM\_OBJHANDLE \* *cseg*, UEM\_CMD\_SEG \* *cseg\_data*)

Создание командного сегмента в ОЗУ виртуального КШ.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bc* | Дескриптор виртуального КШ. |
| out | *cseg* | Дескриптор объекта командного сегмента в ОЗУ КШ. |
| in | *cseg\_data* | Образ командного сегмента в ОЗУ ПЭВМ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

Командный сегмент создается с неустановленной (нулевой) паузой и без внесенных ошибок.

#### ViStatus uem\_cseg\_read (UEM\_OBJHANDLE *cseg*, UEM\_CMD\_SEG \* *cseg\_data*)

Чтение командного сегмента.

Считывание командного сегмента из ОЗУ КШ обратно в ОЗУ ЭВМ.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *cseg* | Дескриптор объекта командного сегмента в ОЗУ КШ. |
| out | *cseg\_data* | Образ командного сегмента в ОЗУ ПЭВМ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_cseg\_gap\_set (UEM\_OBJHANDLE *cseg*, UEM\_WORD *gap*, UEM\_WORD *gap\_flags*, UEM\_WORD *gap\_timeout*)

Программирование паузы перед сообщением.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *cseg* | Дескриптор объекта командного сегмента в ОЗУ КШ. |
| in | *gap* | Значение паузы, в единицах по 0,25 мкс, 0-65535 (См. **Диапазоны значений**). |
| in | *gap\_flags* | Флаги, определяющие способ отсчета паузы, комбинация бит **UEM\_CSEG\_GAP\_XXXX** (Флаги отсчета паузы). |
| in | *gap\_timeout* | Таймаут отсчета паузы (когда отсчет зависит от внешних событий), в мкс, 0-1023 (См. **Диапазоны значений**). |

##### Возвращает:

Код завершения. См. **Коды завершения**.

Для сообщений, для которых не задана явно пауза при помощи данной функции, автоматически будет рассчитана и установлена стандартная пауза в функции **uem\_bcp\_append\_cseg()** или **uem\_bcp\_install()**.

#### ViStatus uem\_cseg\_gap\_get (UEM\_OBJHANDLE *cseg*, UEM\_WORD \* *gap*, UEM\_WORD \* *gap\_flags*, UEM\_WORD \* *gap\_timeout*)

Считывание паузы перед сообщением.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *cseg* | Дескриптор объекта командного сегмента в ОЗУ КШ. |
| out | *gap* | Значение паузы, в единицах по 0,25 мкс, 0-65535. |
| out | *gap\_flags* | Флаги, определяющие способ отсчета паузы. Комбинация бит **UEM\_CSEG\_GAP\_XXXX**. |
| out | *gap\_timeout* | Таймаут отсчета паузы (когда отсчет зависит от внешних событий), в мкс, 0-1023. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_cseg\_gap\_reset (UEM\_OBJHANDLE *cseg*)

Сброс паузы перед сообщением.

Пауза перед сообщением сбрасывается в стандартное (нулевое) значение и помечается как неустановленная (как если бы функция unm\_cseg\_gap\_set() к данному сообщению не применялась). Эта отметка влияет на автоматический расчет паузы в функции **uem\_bcp\_append\_cseg()**, **uem\_bcp\_install()**.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *cseg* | Дескриптор объекта командного сегмента в ОЗУ КШ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_cseg\_word\_gap\_set (UEM\_OBJHANDLE *cseg*, UEM\_WORD *wordnumber*, UEM\_WORD *gap*, UEM\_WORD *gap\_flags*, UEM\_WORD *gap\_timeout*)

Программирование паузы между словами.

Функция устанавливает паузу перед указанным словом командного сегмента. Функция полностью аналогична функции **uem\_cseg\_gap\_set()**, но позволяет установить паузу не перед первым, а перед любым словом командного сегмента. Единственная причина использовать данную функцию - внесение ошибки типа "разрыв сегмента".

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *cseg* | Дескриптор объекта командного сегмента в ОЗУ КШ. |
| in | *wordnumber* | Номер слова, перед которым устанавливается пауза. Нумерация с 0, сквозная, сначала все КС, потом все СД. |
| in | *gap* | Значение паузы, в единицах по 0,25 мкс, 0-65535. (См. **Диапазоны значений**). |
| in | *gap\_flags* | Флаги, определяющие способ отсчета паузы, комбинация бит **Флаги отсчета паузы**. |
| in | *gap\_timeout* | Таймаут отсчета паузы (когда отсчет зависит от внешних событий), в мкс, 0-1023. (См. **Диапазоны значений**). |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_cseg\_word\_gap\_get (UEM\_OBJHANDLE *cseg*, UEM\_WORD *wordnumber*, UEM\_WORD \* *gap*, UEM\_WORD \* *gap\_flags*, UEM\_WORD \* *gap\_timeout*)

Считывание паузы перед словом.

Считывает параметры паузы, установленные функцией **uem\_cseg\_word\_gap\_set()**.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *cseg* | Дескриптор объекта командного сегмента в ОЗУ КШ. |
| in | *wordnumber* | Номер слова. Нумерация с 0, сквозная, сначала все КС, потом все СД. |
| out | *gap* | Значение паузы, в единицах по 0,25 мкс, 0-65535. |
| out | *gap\_flags* | Флаги, определяющие способ отсчета паузы, комбинация бит **Флаги отсчета паузы**. |
| out | *gap\_timeout* | Таймаут отсчета паузы (когда отсчет зависит от внешних событий), в мкс, 0-1023. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_cseg\_error\_set (UEM\_OBJHANDLE *cseg*, UEM\_WORD *wordnumber*, UEM\_ERROR\_TYPE *error\_type*, ViInt32 *error\_pos*, ViInt32 *error\_param*)

Внесение ошибок кодирования.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *cseg* | Дескриптор объекта командного сегмента в ОЗУ КШ. |
| in | *wordnumber* | Номер слова, в которое вносится ошибка. Нумерация с 0, сквозная, сначала все КС, потом все СД. |
| in | *error\_type* | Тип вносимой ошибки. См. **Типы вносимых ошибок кодирования**. |
| in | *error\_pos* | Позиция ошибки. Интерпретируется в зависимости от типа ошибки. |
| in | *error\_param* | Дополнительный параметр. Интерпретируется в зависимости от типа ошибки. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

Допускается внести ошибки кодирования в несколько слов командного сегмента.

#### ViStatus uem\_cseg\_error\_get (UEM\_OBJHANDLE *cseg*, UEM\_WORD *wordnumber*, UEM\_ERROR\_TYPE \* *error\_type*, ViInt32 \* *error\_pos*, ViInt32 \* *error\_param*)

Считывание внесенных ошибок кодирования.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *cseg* | Дескриптор объекта командного сегмента в ОЗУ КШ. |
| in | *wordnumber* | Номер слова, в которое вносится ошибка. Нумерация с 0, сквозная, сначала все КС, потом все СД. |
| out | *error\_type* | Тип вносимой ошибки. См. **Типы вносимых ошибок кодирования**. |
| out | *error\_pos* | Позиция ошибки. Интерпретируется в зависимости от типа ошибки. |
| out | *error\_param* | Дополнительный параметр. Тип вносимой ошибки. Интерпретируется в зависимости от типа ошибки. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_cseg\_sync\_set (UEM\_OBJHANDLE *cseg*, UEM\_WORD *wordnumber*, UEM\_SYNC *sync*)

Установка типа синхроимпульса.

При создании сегмента тип синхроимпульса для каждого слова уже установлен корректно. Программная установка типа синхроимпульса может использоваться для следующих целей:

назначение командных слов в неформатном сообщении (**UEM\_UNF**),

внесение ошибок типа "неверный синхроимпульс",

в сочетании с внесением ошибок кодирования **UEM\_ERRT\_BAD\_SYNCHRO**, расширяет номенклатуру искаженных форм синхроимпульса.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *cseg* | Дескриптор объекта командного сегмента в ОЗУ КШ. |
| in | *wordnumber* | Номер слова, в котором устанавливается синхроимпульс. Нумерация с 0, сквозная, сначала все КС, потом все СД. |
| in | *sync* | Тип синхроимпульса. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_cseg\_sync\_get (UEM\_OBJHANDLE *cseg*, UEM\_WORD *wordnumber*, UEM\_SYNC \* *sync*)

Считывание типа синхроимпульса.

Функция читает тип синхроимпульса в слове сегмента, установленный при создании сегмента или функцией **uem\_cseg\_sync\_set()**.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *cseg* | Дескриптор объекта командного сегмента в ОЗУ КШ. |
| in | *wordnumber* | Номер слова. Нумерация с 0, сквозная, сначала все КС, потом все СД. |
| out | *sync* | Тип синхроимпульса. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_bc\_gap\_create (UEM\_DEVHANDLE *bc*, UEM\_OBJHANDLE \* *cseg*, UEM\_WORD *gap*, UEM\_WORD *gap\_flags*, UEM\_WORD *gap\_timeout*, UEM\_CHANNEL *ch*)

Создание паузы.

Данная функция создает командный сегмент особого вида, не содержащий передаваемых слов, а задающий только паузу в передаче командных сегментов. Аргументы функции аналогичны аргументам функции **uem\_cseg\_gap\_set()**. Их можно прочитать обратно в ОЗУ управляющей ПЭВМ функцией **uem\_cseg\_gap\_get()**. Командный сегмент "пауза" нельзя прочитать функцией **uem\_cseg\_read()**.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bc* | Дескриптор виртуального КШ. |
| out | *cseg* | Дескриптор объекта командного сегмента в ОЗУ КШ. |
| in | *gap* | Значение паузы, в единицах по 0,25 мкс, 0-65535. (См. **Диапазоны значений**). |
| in | *gap\_flags* | Флаги, определяющие способ отсчета паузы, комбинация бит **UEM\_CSEG\_GAP\_XXXX**. |
| in | *gap\_timeout* | Таймаут отсчета паузы (когда отсчет зависит от внешних событий), в мкс, 0-1023. (См. **Диапазоны значений**). |
| in | *ch* | Выбор шины (А/Б). Привязка к шине влияет на интерпретацию флагов в параметре gap\_flags, а также на интерпретацию флагов gap\_flags в следующем сообщении в кадре. В случаях, когда это не важно или выбор неочевиден, рекомендуется указывать **UEM\_CH\_A**. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_bc\_cseg\_overlay (UEM\_DEVHANDLE *bc*, UEM\_OBJHANDLE \* *cseg\_o*, UEM\_OBJHANDLE *cseg\_1*, UEM\_WORD *gap*, UEM\_OBJHANDLE *cseg\_2*)

Создание сообщения с наложением.

Сообщение с наложением - это ситуация, когда два сообщения передаются в шины А и Б одновременно или почти одновременно.

Такие ситуации необходимы для проверки функции вытеснения сообщений в ОУ.

В УЭМ для выполнения такой передачи создается комбинированный командный сегмент, состоящий из двух командных сегментов (по шинам А и Б).

Данная функция создает сегмент с наложением из двух исходных командных сегментов.

Исходные командные сегменты должны быть привязаны к разным шинам.

Создание командного сегмента с наложением обладает следующими особенностями:

Исходные командные сегменты никак не изменяются.

Внесенные ошибки кодирования и смены синхроимпульса переносятся из исходных командных сегментов в новый без изменений.

Паузы, заданные для исходных командных сегментов, игнорируются; в новом командном сегменте паузы устанавливаются в значения по умолчанию.

После создания командного сегмента с наложением исходные командные сегменты можно использовать независимо от нового, или - если они больше не нужны - уничтожить.

Ошибки кодирования и синхроимпульса можно вносить как до построения нового сегмента, так и после - в новый сегмент. Но удобнее делать это до, так как соответствие слов исходных и нового сегмента рассчитывается по специальному алгоритму и не очевидно.

Командный сегмент с наложением нельзя прочитать функцией **uem\_cseg\_read()**.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bc* | Дескриптор виртуального КШ. |
| out | *cseg\_o* | Дескриптор объекта нового командного сегмента в ОЗУ КШ. |
| in | *cseg\_1* | Дескриптор 1-го командного сегмента. |
| in | *gap* | Пауза между началом передачи 1-го командного сегмента и началом передачи 2-го командного сегмента. В единицах по 0,25 мкс. |
| in | *cseg\_2* | Дескриптор 2-го командного сегмента. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_cseg\_type (UEM\_OBJHANDLE *cseg*, UEM\_CSEG\_TYPE \* *type*)

Запрос типа командного сегмента.

При помощи этой функции можно отличить нормальный командный сегмент от специального сегмента - паузы или сегмента с наложением.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *cseg* | Дескриптор объекта командного сегмента. |
| out | *type* | Тип командного сегмента. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_cseg\_desrtoy (UEM\_OBJHANDLE *cseg*)

Уничтожение командного сегмента.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *cseg* | Дескриптор объекта командного сегмента в ОЗУ КШ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

## Создание и настройка кадров и программы КШ

Функции создания и настройки кадров и программ КШ.

### Функции

* ViStatus **uem\_bcp\_create** (**UEM\_OBJHANDLE** \*bcprog, **UEM\_DWORD** max\_size, **UEM\_DEVHANDLE** bc)

*Создание программы КШ.*

* ViStatus **uem\_bcp\_append\_frame** (**UEM\_OBJHANDLE** bcprog, **UEM\_WORD** repeat\_count, **UEM\_WORD** frame\_flags, int \*frameindex)

*Добавление кадра в конец программы КШ.*

* ViStatus **uem\_bcp\_append\_cseg** (**UEM\_OBJHANDLE** bcprog, **UEM\_OBJHANDLE** cseg, int \*csegindex)

*Добавление командного сегмента в конец кадра.*

* ViStatus **uem\_bcp\_discover\_cseg** (**UEM\_OBJHANDLE** bcprog, int frameindex, int csegindex, **UEM\_OBJHANDLE** \*cseg)

*Выяснение командного сегмента.*

* ViStatus **uem\_bcp\_replace\_cseg** (**UEM\_OBJHANDLE** bcprog, int frameindex, int csegindex, **UEM\_OBJHANDLE** cseg)

*Замена командного сегмента в кадре.*

* ViStatus **uem\_bcp\_dimension** (**UEM\_OBJHANDLE** bcprog, int frameindex, int \*dim)

*Запрос размерностей программы КШ.*

* ViStatus **uem\_bcp\_inspect\_frame** (**UEM\_OBJHANDLE** bcprog, int frameindex, **UEM\_WORD** \*repeat\_count, **UEM\_WORD** \*frame\_flags)

*Запрос характеристик кадра.*

* ViStatus **uem\_bcp\_install** (**UEM\_OBJHANDLE** bcprog)

*Установка программы КШ в качестве исполняемой.*

* ViStatus **uem\_bcp\_desrtoy** (**UEM\_OBJHANDLE** bcprog)

*Уничтожение объекта "программа КШ" в ОЗУ КШ.*

* ViStatus **uem\_bcp\_set\_standard\_gaps** (**UEM\_OBJHANDLE** bcprog)

*Расчет и установка стандартных пауз между сообщениями (необязательно).*

### Константы для числа повторов кадра

Данные константы могут использоваться в аргументе **repeat\_count** функции **uem\_bcp\_append\_frame()**.

* #define **UEM\_FRAME\_REPEAT\_UNLIM**  0

*Неограниченное число повторов кадра.*

* #define **UEM\_FRAME\_REPEAT\_MIN**  1

*Минимальное число повторов кадра.*

* #define **UEM\_FRAME\_REPEAT\_MAX**  1023

*Максимальное число повторов кадра.*

* #define **UEM\_FRAME\_REPEAT\_DEFAULT**  (**UEM\_FRAME\_REPEAT\_MIN**)

*Число повторов кадра по умолчанию (1).*

### Флаги кадра

Данные константы могут использоваться в аргументе **frame\_flags** функции **uem\_bcp\_append\_frame()**.

* #define **UEM\_FRAME\_STOP**  0x0001

*Остановка КШ.*

* #define **UEM\_FRAME\_ALLRPT**  0x0002

*Зацикливание программы КШ.*

* #define **UEM\_FRAME\_NONE**  0

*Нет указаний.*

* #define **UEM\_FRAME\_CONT**  (**UEM\_FRAME\_NONE**)

*Продолжение программы КШ.*

* #define **UEM\_FRAME\_DEFAULT**  (**UEM\_FRAME\_STOP**)

*Флаги кадра по умолчанию: остановка КШ.*

### Виды размерностей программы КШ

* #define **UEM\_BCP\_NFRAMES**  (-1)

*Число кадров.*

* #define **UEM\_BCP\_CUR\_SIZE**  (-2)

*Текущий размер.*

* #define **UEM\_BCP\_MAX\_SIZE**  (-3)

*Максимальный размер.*

### Подробное описание

Функции создания и настройки кадров и программ КШ.

Программа КШ (bcp) состоит из кадров, а кадры - из командных сегментов (cseg).

Программа определяет состав и порядок передаваемых сообщений, в том числе - повторы последовательностей сообщений.

В данном разделе описаны функции создания и настройки кадров и программ КШ.

### Макросы

#### #define UEM\_FRAME\_REPEAT\_UNLIM  0

Неограниченное число повторов кадра.

#### #define UEM\_FRAME\_REPEAT\_MIN  1

Минимальное число повторов кадра.

#### #define UEM\_FRAME\_REPEAT\_MAX  1023

Максимальное число повторов кадра.

#### #define UEM\_FRAME\_REPEAT\_DEFAULT  (UEM\_FRAME\_REPEAT\_MIN)

Число повторов кадра по умолчанию (1).

#### #define UEM\_FRAME\_STOP  0x0001

Остановка КШ.

Единичное значение этого флага соответствует указанию остановки работы КШ после передачи текущего кадра (включая выполнение заданного ограниченного (не «бесконечного») количества повторений). Если задано неограниченное число повторений, то при bc\_stop = 1 при первом чтении данного описателя аппаратурой блок обработан не будет и КШ завершит работу. Таким образом, данный флаг не должен указываться, если требуется неограниченное число повторов кадра.

#### #define UEM\_FRAME\_ALLRPT  0x0002

Зацикливание программы КШ.

Единичное значение этого флага, если флаг **UEM\_FRAME\_STOP** установлен в «0», соответствует указанию после завершения передачи текущего кадра (включая выполнение заданного ограниченного (не «бесконечного») количества повторений), перейти к выполнению первого кадра данной программы КШ. Если флаг **UEM\_FRAME\_STOP** установлен в «1», будут выполнены оба действия: остановка КШ (приостановка) и переход к выполнению первого кадра программы, - такая комбинация может быть полезна для организации периодического повторения программы КШ по внешнему синхросигналу.

#### #define UEM\_FRAME\_NONE  0

Нет указаний.

#### #define UEM\_FRAME\_CONT  (UEM\_FRAME\_NONE)

Продолжение программы КШ.

Идентификатор введен как более осмысленный синоним идентификатора **UEM\_FRAME\_NONE**, поскольку отсутствие указаний означает как раз продолжение программы КШ, переход к следующему кадру после выполнения данного кадра, с учетом заданных повторов.

#### #define UEM\_FRAME\_DEFAULT  (UEM\_FRAME\_STOP)

Флаги кадра по умолчанию: остановка КШ.

#### #define UEM\_BCP\_NFRAMES  (-1)

Число кадров.

#### #define UEM\_BCP\_CUR\_SIZE  (-2)

Текущий размер.

#### #define UEM\_BCP\_MAX\_SIZE  (-3)

Максимальный размер.

### Функции

#### ViStatus uem\_bcp\_create (UEM\_OBJHANDLE \* *bcprog*, UEM\_DWORD *max\_size*, UEM\_DEVHANDLE *bc*)

Создание программы КШ.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| out | *bcprog* | Дескриптор объекта программы КШ в ОЗУ КШ. |
| in | *max\_size* | Максимальный размер (число вложенных кадров и сообщений). |
| in | *bc* | Дескриптор виртуального КШ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_bcp\_append\_frame (UEM\_OBJHANDLE *bcprog*, UEM\_WORD *repeat\_count*, UEM\_WORD *frame\_flags*, int \* *frameindex*)

Добавление кадра в конец программы КШ.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bcprog* | Дескриптор программы КШ в ОЗУ КШ. |
| in | *repeat\_count* | Число повторов кадра. Значения: 0-1023; значение 0 (**UEM\_FRAME\_REPEAT\_UNLIM**) означает неограниченное число повторов; см. также константы **UEM\_FRAME\_REPEAT\_XXXX**. |
| in | *frame\_flags* | Управляющие флаги кадра, константы **UEM\_FRAME\_XXXX**. |
| out | *frameindex* | Если этот указатель не равен NULL, в переменную, на которую он указывает, будет записан номер (позиция) нового кадра в программе КШ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_bcp\_append\_cseg (UEM\_OBJHANDLE *bcprog*, UEM\_OBJHANDLE *cseg*, int \* *csegindex*)

Добавление командного сегмента в конец кадра.

Добавление всегда выполняется в конец последнего кадра указанной программы КШ.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bcprog* | Дескриптор программы КШ в ОЗУ КШ. |
| in | *cseg* | Дескриптор командного сегмента в ОЗУ КШ. |
| out | *csegindex* | Если этот указатель не равен NULL, в переменную, на которую он указывает, будет записан номер (позиция) сегмента в кадре. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

Если для добавляемого командного сегмента не запрограммирована пауза перед ним (при помощи **uem\_cseg\_gap\_set()** или другим способом), при добавлении будет рассчитана и установлена стандартная пауза (за исключением командных сегментов, являющихся первыми в кадре: для них стандартная пауза будет установлена функцией **uem\_bcp\_install()**; см. также **uem\_bcp\_set\_standard\_gaps()**).

Допускается добавлять один командный сегмент несколько раз в один и тот же кадр или в разные кадры одной и той же программы КШ или в разные программы КШ в пределах одного виртуального КШ. Однако следует иметь в виду, что установленная или автоматически рассчитанная пауза перед сообщением будет отсчитываться перед каждым вхождением этого сообщения.

#### ViStatus uem\_bcp\_discover\_cseg (UEM\_OBJHANDLE *bcprog*, int *frameindex*, int *csegindex*, UEM\_OBJHANDLE \* *cseg*)

Выяснение командного сегмента.

Функция определяет, какой командный сегмент установлен в указанной позиции указанной программы КШ.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bcprog* | Дескриптор программы КШ в ОЗУ КШ. |
| in | *frameindex* | Номер (позиция) кадра в программе КШ, нумерация с 0. |
| in | *csegindex* | Номер (позиция) командного сегмента в кадре, нумерация с 0. |
| out | *cseg* | Дескриптор командного сегмента в ОЗУ КШ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_bcp\_replace\_cseg (UEM\_OBJHANDLE *bcprog*, int *frameindex*, int *csegindex*, UEM\_OBJHANDLE *cseg*)

Замена командного сегмента в кадре.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bcprog* | Дескриптор программы КШ в ОЗУ КШ. |
| in | *frameindex* | Номер (позиция) кадра в программе КШ, нумерация с 0. |
| in | *csegindex* | Номер (позиция) командного сегмента в кадре, нумерация с 0. |
| in | *cseg* | Дескриптор командного сегмента в ОЗУ КШ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

Если для вставляемого командного сегмента не запрограммирована пауза перед ним (при помощи **uem\_cseg\_gap\_set()** или другим способом), при добавлении будет рассчитана и установлена стандартная пауза (за исключением командных сегментов, являющихся первыми в кадре: для них стандартная пауза будет установлена функцией **uem\_bcp\_install()**; см. также **uem\_bcp\_set\_standard\_gaps()**).

Допускается вставлять один командный сегмент несколько раз в один и тот же кадр или в разные кадры одной и той же программы КШ или в разные программы КШ в пределах одного виртуального КШ. Однако следует иметь в виду, что установленная или автоматически рассчитанная пауза перед сообщением будет отсчитываться перед каждым вхождением этого сообщения.

#### ViStatus uem\_bcp\_dimension (UEM\_OBJHANDLE *bcprog*, int *frameindex*, int \* *dim*)

Запрос размерностей программы КШ.

Функция возвращает число командных сегментов в указанном кадре программы, а также - показатели размерности самой программы.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bcprog* | Дескриптор программы КШ в ОЗУ КШ. |
| in | *frameindex* | Значения >= 0: Номер (позиция) кадра в программы КШ, нумерация с 0; значения < 0: запрос размерностей самой программы КШ, допускается указать одну из констант **UEM\_BCP\_XXX**. |
| out | *dim* | В эту переменную записывается значение запрошенной размерности: при frameindex >= 0 возвращается число командных сегментов в кадре с номером frameindex, при frameindex < 0 возвращается число кадров в программе КШ, текущий размер программы или максимальный размер программы, в зависимости от значения frameindex (см. **UEM\_BCP\_XXX**). |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_bcp\_inspect\_frame (UEM\_OBJHANDLE *bcprog*, int *frameindex*, UEM\_WORD \* *repeat\_count*, UEM\_WORD \* *frame\_flags*)

Запрос характеристик кадра.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bcprog* | Дескриптор программы КШ в ОЗУ КШ. |
| in | *frameindex* | Номер (позиция) кадра в программе КШ, нумерация с 0. |
| out | *repeat\_count* | Число повторов кадра. Значения: 0-1023; значение 0 (**UEM\_FRAME\_REPEAT\_UNLIM**) означает неограниченное число повторов; см. также константы **UEM\_FRAME\_REPEAT\_XXXX**. |
| out | *frame\_flags* | Управляющие флаги кадра, константы **UEM\_FRAME\_XXXX**. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_bcp\_install (UEM\_OBJHANDLE *bcprog*)

Установка программы КШ в качестве исполняемой.

Установка заключается в загрузке адреса этой программы в соответствующий управляющий регистр УЭМ.

Перед установкой программы функция убеждается, что для всех командных сегментов в программе КШ запрограммированы паузы перед ними, и устанавливает стандартные паузы для тех сегментов, для которых они не запрограммированы, путем обращения к функции **uem\_bcp\_set\_standard\_gaps()** (см.).

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bcprog* | Дескриптор программы КШ в ОЗУ КШ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_bcp\_desrtoy (UEM\_OBJHANDLE *bcprog*)

Уничтожение объекта "программа КШ" в ОЗУ КШ.

Уничтожение программы КШ не уничтожает включенные в нее командные сегменты.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bcprog* | Дескриптор программы КШ в ОЗУ КШ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_bcp\_set\_standard\_gaps (UEM\_OBJHANDLE *bcprog*)

Расчет и установка стандартных пауз между сообщениями (необязательно).

Данная функция рассчитывает и устанавливает стандартные паузы между сообщениями, включенными в программу КШ, за исключением тех сообщений, для которых паузы уже были установлены ранее при помощи **uem\_cseg\_gap\_set()**, данной функции, или другим способом.

Обращение к данной функции необязательно, так как она вызывается автоматически из функции **uem\_bcp\_install()**.

Стандартная пауза устанавливается как интервал времени между началом передачи предыдущего командного сегмента и началом передачи данного командного сегмента. При расчете паузы учитывается продолжительность командного и ответных сегментов предыдущего сообщения, таймаут (таймауты) ответа **UEM\_RTMO** и время минимальной паузы между сообщениями **UEM\_MIN\_T2**. Таким образом, устанавливается минимальная пауза, обеспечивающая передачу данных без нарушений протокола, при условии, что ОУ также не нарушает протокол.

Для первого сообщения зацикленного кадра учитывается как сообщение, предшествующее по порядку сообщений в программе КШ, так и сообщение, предшествующее при повторах.

Возможен конфликт выбора шины для отсчета паузы в первом сообщения зацикленного кадра. Данная функция в этом случае отдает предпочтение сообщению, предшествующему при повторах. Для разрешения конфликта рекомендуется в этом случае задавать паузу и правила ее отсчета самостоятельно при помощи **uem\_cseg\_gap\_set()** или **uem\_bc\_gap\_create()**.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bcprog* | Дескриптор программы КШ в ОЗУ КШ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

## Запуск и остановка КШ

Порядок запуска, остановки и контроля активности КШ.

Запуск виртуального КШ выполняется функциями **uem\_start()** или **uem\_bc\_start()**, а остановка - функциями **uem\_stop()** или **uem\_bc\_stop()**.

В зависимости от программы КШ, КШ также может останавливаться самостоятельно, при исчерпании заданной последовательности сообщений. Для проверки состояния активности КШ используются функции **uem\_is\_running()** или **uem\_bc\_state\_ex()**.

Переходы расширенного состояния КШ приведены в следующей таблице:

| Исходное состояние | Событие | Состояние после события |
| --- | --- | --- |
| **UEM\_BC\_STOPPED** | **uem\_start()** | **UEM\_BC\_RUNNING** |
| **UEM\_BC\_STOPPED** | **uem\_bc\_start()** с параметром **flags** = **UEM\_BC\_START\_NOW** | **UEM\_BC\_RUNNING** |
| **UEM\_BC\_STOPPED** | **uem\_bc\_start()** с параметром **flags** = **UEM\_BC\_START\_WAITING** | **UEM\_BC\_WAITING** |
| **UEM\_BC\_WAITING** | **uem\_start()** | **UEM\_BC\_RUNNING** |
| **UEM\_BC\_WAITING** | **uem\_bc\_start()** с параметром **flags** = **UEM\_BC\_START\_NOW** | **UEM\_BC\_RUNNING** |
| **UEM\_BC\_WAITING** | Сигнал **sync\_in\_2** или его внутренняя имитация | **UEM\_BC\_RUNNING** |
| **UEM\_BC\_WAITING** | Получение ОУ команды управления "Принять управление интерфейсом", когда в ответном сегменте установлен флаг **UEM\_RTDES\_DBCA\_BCSTART** | **UEM\_BC\_RUNNING** |
| **UEM\_BC\_WAITING** | **uem\_stop()** | **UEM\_BC\_STOPPED** |
| **UEM\_BC\_WAITING** | **uem\_bc\_stop()** с параметром **flags** = **UEM\_BC\_STOP\_NOW** | **UEM\_BC\_STOPPED** |
| **UEM\_BC\_RUNNING** | **uem\_stop()** | **UEM\_BC\_STOPPED** |
| **UEM\_BC\_RUNNING** | **uem\_bc\_stop()** с параметром **flags** = **UEM\_BC\_STOP\_NOW** | **UEM\_BC\_STOPPED** |
| **UEM\_BC\_RUNNING** | **uem\_bc\_stop()** с параметром **flags** = **UEM\_BC\_STOP\_ON\_FRAME** | **UEM\_BC\_RUNNING**, ожидание завершения кадра |
| **UEM\_BC\_RUNNING** | Ожидаемое завершение кадра | **UEM\_BC\_WAITING** |
| **UEM\_BC\_RUNNING** | Завершение программы КШ (отработка флага кадра **UEM\_FRAME\_STOP**) | **UEM\_BC\_WAITING** |

## Передача сообщений

Описание функции передачи отдельного сообщения.

### Функции

* ViStatus **uem\_bc\_send\_receive** (**UEM\_DEVHANDLE** bc, **UEM\_CMD\_SEG** \*cseg\_data, **UEM\_BM\_MESSAGE** \*\*msg\_and\_resp)

*Передача отдельного сообщения и получение ответа на него.*

### Подробное описание

Описание функции передачи отдельного сообщения.

### Функции

#### ViStatus uem\_bc\_send\_receive (UEM\_DEVHANDLE *bc*, UEM\_CMD\_SEG \* *cseg\_data*, UEM\_BM\_MESSAGE \*\* *msg\_and\_resp*)

Передача отдельного сообщения и получение ответа на него.

Данная функция не требует предварительного конфигурирования КШ путем создания командных сегментов, кадров и программы КШ.

Для выполнения этого действия используются виртуальные КШ и МШ в составе УЭМ. Следует иметь в виду, что функция их переконфигурирует. Открывать МШ для выполнения данной функции не обязательно.

В сообщениях, отправляемых таким образом, нельзя регулировать паузы и вносить ошибки кодирования.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bc* | Дескриптор виртуального КШ. |
| in | *cseg\_data* | Образ командного сегмента, который надо передать. |
| out | *msg\_and\_resp* | Переданный командный сегмент и ответ на него, по данным МШ. В эту переменную записывается указатель на структуру **UEM\_BM\_MESSAGE**. Память для этой структуры выделяется динамически, и приложение ответственно за ее освобождение. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

## Функции ОУ

Описания функций оконечного устройства.

### Функции

* ViStatus **uem\_response\_create** (**UEM\_OBJHANDLE** \*resp, **UEM\_DEVHANDLE** rt, **UEM\_DWORD** rtdes, **UEM\_DWORD** illeg\_mask, **UEM\_WORD** status, **UEM\_WORD** ndatawords, **UEM\_WORD** \*data)

*Создание ответного сегмента.*

* ViStatus **uem\_response\_read** (**UEM\_OBJHANDLE** resp, **UEM\_DWORD** \*rtdes, **UEM\_DWORD** \*illeg\_mask, **UEM\_WORD** \*status, **UEM\_WORD** \*ndatawords, **UEM\_WORD** \*data)

*Считывание ответного сегмента.*

* ViStatus **uem\_response\_gap\_set** (**UEM\_OBJHANDLE** resp, **UEM\_WORD** gap)

*Установка паузы перед передачей ответного сегмента.*

* ViStatus **uem\_response\_gap\_get** (**UEM\_OBJHANDLE** resp, **UEM\_WORD** \*gap)

*Считывание паузы перед передачей ответного сегмента.*

* ViStatus **uem\_response\_word\_gap\_set** (**UEM\_OBJHANDLE** resp, **UEM\_WORD** wordnumber, **UEM\_WORD** gap)

*Установка паузы перед передачей слова ответного сегмента.*

* ViStatus **uem\_response\_word\_gap\_get** (**UEM\_OBJHANDLE** resp, **UEM\_WORD** wordnumber, **UEM\_WORD** \*gap)

*Считывание паузы перед передачей слова ответного сегмента.*

* ViStatus **uem\_response\_error\_set** (**UEM\_OBJHANDLE** resp, **UEM\_WORD** wordnumber, **UEM\_ERROR\_TYPE** error\_type, ViInt32 error\_pos, ViInt32 error\_param)

*Внесение ошибок кодирования в ответный сегмент.*

* ViStatus **uem\_response\_error\_get** (**UEM\_OBJHANDLE** resp, **UEM\_WORD** wordnumber, **UEM\_ERROR\_TYPE** \*error\_type, ViInt32 \*error\_pos, ViInt32 \*error\_param)

*Считывание ошибок кодирования из ответного сегмента.*

* ViStatus **uem\_response\_sync\_set** (**UEM\_OBJHANDLE** resp, **UEM\_WORD** wordnumber, **UEM\_SYNC** sync)

*Установка типа синхроимпульса.*

* ViStatus **uem\_response\_sync\_get** (**UEM\_OBJHANDLE** resp, **UEM\_WORD** wordnumber, **UEM\_SYNC** \*sync)

*Считывание типа синхроимпульса.*

* ViStatus **uem\_rt\_install\_response** (**UEM\_DEVHANDLE** rt, **UEM\_BOOL** transmit, **UEM\_WORD** sa, **UEM\_OBJHANDLE** resp)

*Установка ответного сегмента как ответа на команду передачи данных.*

* ViStatus **uem\_rt\_install\_response\_MODE** (**UEM\_DEVHANDLE** rt, **UEM\_BOOL** transmit, **UEM\_WORD** mode, **UEM\_WORD** modecode, **UEM\_OBJHANDLE** resp)

*Установка ответного сегмента как ответа на команду управления.*

* ViStatus **uem\_rt\_discover\_response** (**UEM\_DEVHANDLE** rt, **UEM\_BOOL** transmit, **UEM\_WORD** sa, **UEM\_OBJHANDLE** \*resp)

*Выяснение ответа на команду передачи данных.*

* ViStatus **uem\_rt\_discover\_response\_MODE** (**UEM\_DEVHANDLE** rt, **UEM\_BOOL** transmit, **UEM\_WORD** mode, **UEM\_WORD** modecode, **UEM\_OBJHANDLE** \*resp)

*Выяснение ответа на команду управления.*

* ViStatus **uem\_response\_destroy** (**UEM\_OBJHANDLE** resp)

*Уничтожение объекта ответного сегмента в ОЗУ ОУ.*

### Признаки обработки командного слова в ОУ

Из объединения (по |) этих констант составляется аргумент **rtdes** в функциях **uem\_response\_create()**, **uem\_response\_read()**.

* #define **UEM\_RTDES\_SW\_DIS**  (1<<15)

*Не отвечать.*

* #define **UEM\_RTDES\_DBCA**  (1<<6)

*Принять управление интерфейсом.*

* #define **UEM\_RTDES\_DBCA\_BCSTART**  (1<<7)

*Запустить КШ.*

* #define **UEM\_RTDES\_COM\_ILLEGAL**  (1<<17)

*Недопустимая команда.*

* #define **UEM\_RTDES\_LCMD\_DW**  (1<<13)

*Передать последнюю команду.*

* #define **UEM\_RTDES\_SWB\_SAV**  (1<<12)

*Автоматическое формирование флагов ОС.*

* #define **UEM\_RTDES\_WRONG\_CH**  (1<<14)

*Отвечать по другой шине.*

* #define **UEM\_RTDES\_WA**  (1<<8)

*Циркулярный возврат данных.*

* #define **UEM\_RTDES\_WA\_BRCST**  (1<<31)

*Циркулярный возврат в групповых командах.*

* #define **UEM\_RTDES\_ILLEG\_MASK**  (1<<16)

*Задать маску допустимых команд в зависимости от количества слов данных.*

* #define **UEM\_RTDES\_DEFAULT**  (**UEM\_RTDES\_SWB\_SAV**)

*Значение параметра* ***rtdes*** *по умолчанию.*

### Подробное описание

Описания функций оконечного устройства.

Для использования этих функций необходимо получить дескриптор виртуального ОУ при помощи функции **uem\_rt\_init()**.

Конфигурирование ОУ заключается в создании и настройке ответных сегментов и установке этих сегментов в качестве ответов на заданные командные слова МКПД. (См. **Командные и ответные сегменты**.)

Для того чтобы ОУ отвечал на командные слова в соответствии с установленными ответами, его необходимо запустить в работу при помощи функции **uem\_start()**.

### Макросы

#### #define UEM\_RTDES\_SW\_DIS  (1<<15)

Не отвечать.

Если флаг установлен (UEM\_RTDES\_SW\_DIS=1), то ОС и СД (если они есть) не передаются в МКПД. При установке данного флага остальные флаги игнорируются.

#### #define UEM\_RTDES\_DBCA  (1<<6)

Принять управление интерфейсом.

Если флаг установлен (UEM\_RTDES\_DBCA=1), то вне зависимости от кода КС и соответствующего формата сообщения, в ОС устанавливается признак **Принято управление интерфейсом** (если установлен флаг **UEM\_RTDES\_SWB\_SAV**), СД не передаются, после завершения передачи ОС данное виртуальное устройство ОУ отключается. Возможное использование данного флага – поддержка возможности обработки КУ «Принять управление интерфейсом», если данная КУ применима для выбранного адреса ОУ.

#### #define UEM\_RTDES\_DBCA\_BCSTART  (1<<7)

Запустить КШ.

Флаг значим только при установленном флаге **UEM\_RTDES\_DBCA**. Если флаг установлен (UEM\_RTDES\_DBCA\_BCSTART=1), и если КШ в момент передачи текущего ОС находится в состоянии ожидания, то КШ запускается. Возможное использование данного флага – поддержка возможности обработки КУ «Принять управление интерфейсом» с передачей управления собственному КШ, если данная КУ применима для выбранного адреса ОУ.

#### #define UEM\_RTDES\_COM\_ILLEGAL  (1<<17)

Недопустимая команда.

Флаг устанавливает признак недопустимости ОС для всех КС, для которых установлен данный ответ. В этом случае в ОС устанавливается бит статуса **ОШС**, слова данных в ответе не передаются. При сброшенном флаге КС считаются допустимыми. Допустимость/недопустимость в зависимости от числа слов данных можно установить в параметре **illeg\_mask** функции **uem\_response\_create()** .

#### #define UEM\_RTDES\_LCMD\_DW  (1<<13)

Передать последнюю команду.

При установленном флаге (UEM\_RTDES\_LCMD\_DW=1) и не установленных флагах **UEM\_RTDES\_SW\_DIS**, **UEM\_RTDES\_DBCA** после передачи ОС, вне зависимости от значения параметра **ndatawords** функции **uem\_response\_create()**, передается одно СД, содержащее код последней предшествующей достоверной команды к данному адресу ОУ. Исключением является код команды управления «Передать последнюю команду». Возможное использование данного флага – поддержка возможности обработки КУ «Передать последнюю команду».

**Примечание:** в данном случае будут изменены только информационные разряды первого СД. Все остальные параметры должны заполняться корректно по общим правилам. В частности, ответный сегмент должен содержать как минимум одно слово данных. Внесенные в это слово ошибки кодирования, синхроимпульса, паузы будут отработаны оборудованием.

#### #define UEM\_RTDES\_SWB\_SAV  (1<<12)

Автоматическое формирование флагов ОС.

Флаг значим только при не установленном флаге **UEM\_RTDES\_SW\_DIS** и определяет правила формирования флагов ОС, которые приведены в **таблице 3**.

Таблица 3. **Правила формирования признаков ОС**

| **Флаги ОС** | **UEM\_RTDES\_SWB\_SAV=0**  (прямое задание значений разрядов ОС) | **UEM\_RTDES\_SWB\_SAV=1**  (автоматическое формирование значений разрядов ОС в зависимости от контекста предыдущих сообщений в МКПД) |
| --- | --- | --- |
| «Адрес ОУ» | Определяется разрядами [15:11] параметра **status** функции **uem\_response\_create()** | Определяется адресом ОУ в КС |
| «Ошибка в сообщении» | Определяется разрядом [10] параметра **status** функции **uem\_response\_create()** | Определяется исходя из значения флагов **UEM\_RTDES\_COM\_ILLEGAL**, параметра **illeg\_mask** функции **uem\_response\_create()** и таблицы допустимости (настройка допустимости имеет приоритет), состояния достоверности и допустимости предыдущего КС, с учетом правил ГОСТ Р 52070 для КУ «Передать ОС» и «Передать последнюю команду» |
| «Передача ОС» | Определяется разрядом [9] параметра **status** функции **uem\_response\_create()** | Нулевое значение |
| «Запрос на обслуживание» | Определяется разрядом [8] параметра **status** функции **uem\_response\_create()** | Определяется разрядом [8] параметра **status** функции **uem\_response\_create()** |
| Резервные | Определяются разрядами [7:5] параметра **status** функции **uem\_response\_create()** | Нулевое значение |
| «Принята групповая команда» | Определяется разрядом [3] параметра **status** функции **uem\_response\_create()** | Определяется исходя из состояния и значений текущего и предыдущего КС, устанавливается в «1» в ответ на КУ «Передать ОС» и «Передать последнюю команду» |
| «Абонент занят» | Определяется разрядом [4] параметра **status** функции **uem\_response\_create()** | Определяется разрядом [4] параметра **status** функции **uem\_response\_create()** |
| «Неисправность абонента» | Определяется разрядом [2] параметра **status** функции **uem\_response\_create()** | Определяется разрядом [2] параметра **status** функции **uem\_response\_create()** |
| «Принято управление интерфейсом» | Определяется разрядом [1] параметра **status** функции **uem\_response\_create()** | Определяется значением флага **UEM\_RTDES\_DBCA** |
| «Неисправность ОУ» | Определяется разрядом [0] параметра **status** функции **uem\_response\_create()** | Определяется разрядом [0] параметра **status** функции **uem\_response\_create()** и текущим состоянием блокировки признака «Неисправность ОУ» по соответствующим КУ, если нет установки запрета обработки таких КУ флагом **UEM\_BRTF\_DIS** |

#### #define UEM\_RTDES\_WRONG\_CH  (1<<14)

Отвечать по другой шине.

Если флаг установлен (UEM\_RTDES\_WRONG\_CH=1), то ОС (если его передача предусмотрена) и СД (если они есть) передаются по резервной шине МКПД по отношению к той шине, по которой поступило КС.

#### #define UEM\_RTDES\_WA  (1<<8)

Циркулярный возврат данных.

Флаг использования циркулярного возврата данных, значим только при нулевых значениях флагов **UEM\_RTDES\_SW\_DIS**, **UEM\_DBCA**, **UEM\_RTDES\_LCMD\_DW**, и только в описателях КС информационного обмена (подадреса от 1 до 30 включительно, а также если установлен запрет команд управления, подадреса 0 и 31).

Установка данного флага (UEM\_RTDES\_WA=1) для КС на прием означает, что будут сохранены поступающие после данного КС СД (если они фактически есть) в отдельном (для данного адреса ОУ) буфере данных циркулярного возврата. Установка данного флага (UEM\_RTDES\_WA=1) для КС на передачу означает, что в передаваемых СД (если их передача происходит) будут находиться ранее сохраненные данные из буфера циркулярного возврата.

Рекомендация по включению режима циркулярного возврата для выбранного адреса ОУ.

Установить флаг **UEM\_RTDES\_WA** для одного из КС приема (с нужным значением подадреса приема циркулярного возврата данных) и одного из КС передачи (с нужным значением подадреса передачи циркулярного возврата данных), при необходимости использовать данные групповых сообщений – также для данного адреса ОУ включить разрешение приема слов данных в групповых командах.

ГОСТ Р 52070 рекомендует использовать только подадрес 30 для приема и передачи циркулярного возврата. В общем случае, если флаг **UEM\_RTDES\_WA** установлен для нескольких кодов КС, обрабатываемых данным ОУ, принимаемые данные будут направляться в один и тот же буфер приема, и могут быть переданы из этого буфера в ответ на КС с несколькими подадресами.

#### #define UEM\_RTDES\_WA\_BRCST  (1<<31)

Циркулярный возврат в групповых командах.

Флаг обеспечивает запоминание в буфере слов данных, поступающих в групповых командах (аналогично флагу **UEM\_RTDES\_WA** для негрупповых команд).

#### #define UEM\_RTDES\_ILLEG\_MASK  (1<<16)

Задать маску допустимых команд в зависимости от количества слов данных.

При установленном флаге (UEM\_RTDES\_ILLEG\_MASK = 1) допустимость команд в зависимости указанного в них количества слов данных определяется параметром **illeg\_mask** в функции **uem\_response\_create()**. При сброшенном флаге (UEM\_RTDES\_ILLEG\_MASK = 0) команды считаются допустимыми при любом числе слов данных, параметр **illeg\_mask** игнорируется. Флаг не должен устанавливаться в ответах на команды управления.

#### #define UEM\_RTDES\_DEFAULT  (UEM\_RTDES\_SWB\_SAV)

Значение параметра **rtdes** по умолчанию.

### Функции

#### ViStatus uem\_response\_create (UEM\_OBJHANDLE \* *resp*, UEM\_DEVHANDLE *rt*, UEM\_DWORD *rtdes*, UEM\_DWORD *illeg\_mask*, UEM\_WORD *status*, UEM\_WORD *ndatawords*, UEM\_WORD \* *data*)

Создание ответного сегмента.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| out | *resp* | Дескриптор ответного сегмента в ОЗУ ОУ. |
| in | *rt* | Дескриптор ОУ. |
| in | *rtdes* | Битовая строка признаков правил обработки командного слова в ОУ. Составляется как объединение (по |) констант **UEM\_RTDES\_XXXX**. По умолчанию выставляется **UEM\_RTDES\_DEFAULT**. |
| in | *illeg\_mask* | Битовая маска недопустимости команд в зависимости от числа СД. Данный аргумент используется, если в аргументе **rtdes** установлен бит **UEM\_RTDES\_ILLEG\_MASK** и не установлены биты **UEM\_RTDES\_COM\_ILLEGAL**, **UEM\_RTDES\_SW\_DIS**, в противном случае - игнорируется. Бит с номером n - признак недопустимости командного слова с числом слов данных n, 1 <= n <= 31. Бит с номером 0 - признак недопустимости командного слова с числом слов данных 32. Значение 1 в бите означает, что команда недопустима, значение 0 - допустима. |
| in | *status* | Ответное слово. |
| in | *ndatawords* | Число слов данных. Допустимые значения: 0-63. Аппаратура УЭМ способна формировать до 62 слов данных в ответном сегменте.    Специальная константа 63 (**UEM\_NDATA\_BY\_CW**) указывает, что число слов данных в ответном сегменте определяется полученным командным словом. Фактическое число слов данных в массиве **data** в этом случае должно быть 32. |
| in | *data* | Массив значений слов данных. Длина определяется аргументом **ndatawords** . |

##### Возвращает:

Код завершения. См. **Коды завершения**.

Ответный сегмент создается без внесенных ошибок. Пауза перед ответным сегментом определяется временнЫм параметром **UEM\_MIN\_T1**, заданным для данного виртуального ОУ.

#### ViStatus uem\_response\_read (UEM\_OBJHANDLE *resp*, UEM\_DWORD \* *rtdes*, UEM\_DWORD \* *illeg\_mask*, UEM\_WORD \* *status*, UEM\_WORD \* *ndatawords*, UEM\_WORD \* *data*)

Считывание ответного сегмента.

Функция позволяет прочитать данные ответного сегмента, созданного функцией **uem\_response\_create()**, обратно в ОЗУ управляющей ПЭВМ.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *resp* | Дескриптор ответного сегмента в ОЗУ ОУ. |
| out | *rtdes* | Битовая строка признаков правил обработки командного слова в ОУ. Объединение констант **UEM\_RTDES\_XXXX**. |
| out | *illeg\_mask* | Битовая маска недопустимости команд в зависимости от числа СД. См. **illeg\_mask** в **uem\_response\_create()**. |
| out | *status* | Ответное слово. |
| out | *ndatawords* | Число слов данных. |
| out | *data* | Массив значений слов данных. Длина должна быть не меньше 62 слов. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_response\_gap\_set (UEM\_OBJHANDLE *resp*, UEM\_WORD *gap*)

Установка паузы перед передачей ответного сегмента.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *resp* | Дескриптор ответного сегмента в ОЗУ ОУ. |
| in | *gap* | Значение паузы. Задается в единицах по 0,25 мкс. Допустимо: 0-65535. (См. **Диапазоны значений**). |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_response\_gap\_get (UEM\_OBJHANDLE *resp*, UEM\_WORD \* *gap*)

Считывание паузы перед передачей ответного сегмента.

Функция считывает значение паузы, установленное функцией **uem\_response\_gap\_set()**, либо установленное автоматически в **uem\_response\_create()**.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *resp* | Дескриптор ответного сегмента в ОЗУ ОУ. |
| out | *gap* | Значение паузы. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_response\_word\_gap\_set (UEM\_OBJHANDLE *resp*, UEM\_WORD *wordnumber*, UEM\_WORD *gap*)

Установка паузы перед передачей слова ответного сегмента.

Функция полностью аналогична функции **uem\_response\_gap\_set()**, но позволяет установить паузу не перед первым, а перед любым словом ответного сегмента. Единственная причина использовать эту функцию - для внесения ошибки типа "разрыв сегмента".

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *resp* | Дескриптор ответного сегмента в ОЗУ ОУ. |
| in | *wordnumber* | Номер слова. Нумерация с 0, сквозная, сначала ответное слово, потом слова данных. |
| in | *gap* | Значение паузы. Задается в единицах по 0,25 мкс. Допустимо: 0-65535. (См. **Диапазоны значений**). |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_response\_word\_gap\_get (UEM\_OBJHANDLE *resp*, UEM\_WORD *wordnumber*, UEM\_WORD \* *gap*)

Считывание паузы перед передачей слова ответного сегмента.

Функция считывает значение паузы, установленное функцией **uem\_response\_word\_gap\_set()**.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *resp* | Дескриптор ответного сегмента в ОЗУ ОУ. |
| in | *wordnumber* | Номер слова. Нумерация с 0, сквозная, сначала ответное слово, потом слова данных. |
| out | *gap* | Значение паузы. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_response\_error\_set (UEM\_OBJHANDLE *resp*, UEM\_WORD *wordnumber*, UEM\_ERROR\_TYPE *error\_type*, ViInt32 *error\_pos*, ViInt32 *error\_param*)

Внесение ошибок кодирования в ответный сегмент.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *resp* | Дескриптор ответного сегмента в ОЗУ ОУ. |
| in | *wordnumber* | Номер слова. Нумерация с 0, сквозная, сначала ответное слово, потом слова данных. |
| in | *error\_type* | Тип вносимой ошибки. См. **Типы вносимых ошибок кодирования**. |
| in | *error\_pos* | Позиция ошибки. Интерпретируется в зависимости от типа ошибки. |
| in | *error\_param* | Дополнительный параметр. Интерпретируется в зависимости от типа ошибки. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

Допускается внести ошибки кодирования в несколько слов ответного сегмента.

#### ViStatus uem\_response\_error\_get (UEM\_OBJHANDLE *resp*, UEM\_WORD *wordnumber*, UEM\_ERROR\_TYPE \* *error\_type*, ViInt32 \* *error\_pos*, ViInt32 \* *error\_param*)

Считывание ошибок кодирования из ответного сегмента.

Функция позволяет прочитать данные ошибок кодирования, внесенных функцией **uem\_response\_error\_set()**, обратно в ОЗУ управляющей ПЭВМ.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *resp* | Дескриптор ответного сегмента в ОЗУ ОУ. |
| in | *wordnumber* | Номер слова. Нумерация с 0, сквозная, сначала ответное слово, потом слова данных. |
| out | *error\_type* | Тип вносимой ошибки. |
| out | *error\_pos* | Позиция ошибки. Интерпретируется в зависимости от типа ошибки. |
| out | *error\_param* | Дополнительный параметр. Интерпретируется в зависимости от типа ошибки. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_response\_sync\_set (UEM\_OBJHANDLE *resp*, UEM\_WORD *wordnumber*, UEM\_SYNC *sync*)

Установка типа синхроимпульса.

При создании сегмента тип синхроимпульса для каждого слова уже установлен корректно. Программная установка типа синхроимпульса может использоваться для следующих целей:

внесение ошибок типа "неверный синхроимпульс",

в сочетании с внесением ошибок кодирования **UEM\_ERRT\_BAD\_SYNCHRO** расширяет номенклатуру искаженных форм синхроимпульса.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *resp* | Дескриптор ответного сегмента в ОЗУ ОУ. |
| in | *wordnumber* | Номер слова, в котором устанавливается синхроимпульс. Нумерация с 0, сквозная, сначала ответное слово, потом слова данных. |
| in | *sync* | Тип синхроимпульса. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_response\_sync\_get (UEM\_OBJHANDLE *resp*, UEM\_WORD *wordnumber*, UEM\_SYNC \* *sync*)

Считывание типа синхроимпульса.

Функция читает тип синхроимпульса в слове сегмента, установленный при создании сегмента или функцией **uem\_response\_sync\_set()**.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *resp* | Дескриптор ответного сегмента в ОЗУ ОУ. |
| in | *wordnumber* | Номер слова. Нумерация с 0, сквозная, сначала ответное слово, потом слова данных. |
| out | *sync* | Тип синхроимпульса. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_rt\_install\_response (UEM\_DEVHANDLE *rt*, UEM\_BOOL *transmit*, UEM\_WORD *sa*, UEM\_OBJHANDLE *resp*)

Установка ответного сегмента как ответа на команду передачи данных.

Данная функция устанавливает указанный ответный сегмент в качестве ответа на команду передачи данных.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *rt* | Дескриптор виртуального ОУ. |
| in | *transmit* | Признак Передачи(1)/Приема(0) в командном слове. |
| in | *sa* | Подадрес |
| in | *resp* | Дескриптор ответного сегмента в ОЗУ ОУ. Указанный ответный сегмент будет установлен в качестве ответа на указанное командное слово.  Этот аргумент также может принимать нулевое значение (NULL), которое означает указание деинсталлировать ответ на указанное командное слово, ничем его не заменяя.    В ответ на командные слова, для которых не задан ответный сегмент, автоматически генерируется ответ в виде одного ответного слова с признаком "Ошибка в сообщении".    Допускается устанавливать один и тот же ответный сегмент в качестве ответа на разные команды. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_rt\_install\_response\_MODE (UEM\_DEVHANDLE *rt*, UEM\_BOOL *transmit*, UEM\_WORD *mode*, UEM\_WORD *modecode*, UEM\_OBJHANDLE *resp*)

Установка ответного сегмента как ответа на команду управления.

Данная функция устанавливает указанный ответный сегмент в качестве ответа на команду управления.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *rt* | Дескриптор виртуального ОУ. |
| in | *transmit* | Признак Передачи(1)/Приема(0) в командном слове. |
| in | *mode* | Код режима управления. Допускаются значения 0 (**UEM\_MODE\_0**) и 31 (**UEM\_MODE\_31**). |
| in | *modecode* | Код команды управления. |
| in | *resp* | Дескриптор ответного сегмента в ОЗУ ОУ. Указанный ответный сегмент будет установлен в качестве ответа на указанное командное слово.  Этот аргумент также может принимать нулевое значение (NULL), которое означает указание деинсталлировать ответ на указанное командное слово, ничем его не заменяя.    В ответ на командные слова, для которых не задан ответный сегмент, автоматически генерируется ответ в виде одного ответного слова с признаком "Ошибка в сообщении".    Допускается устанавливать один и тот же ответный сегмент в качестве ответа на разные команды. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_rt\_discover\_response (UEM\_DEVHANDLE *rt*, UEM\_BOOL *transmit*, UEM\_WORD *sa*, UEM\_OBJHANDLE \* *resp*)

Выяснение ответа на команду передачи данных.

Данная функция позволяет выяснить, какой ответный сегмент установлен функцией **uem\_rt\_install\_response()** в качестве ответа на указанную команду. Аргументы полностью аналогичны параметрам **uem\_rt\_install\_response()**.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *rt* | Дескриптор виртуального ОУ. |
| in | *transmit* | Признак Передачи(1)/Приема(0) в командном слове. |
| in | *sa* | Подадрес. |
| out | *resp* | Дескриптор ответного сегмента в ОЗУ ОУ. Если ответ на указанную команду не установлен, в качестве значения в этом аргументе будет возвращен NULL. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_rt\_discover\_response\_MODE (UEM\_DEVHANDLE *rt*, UEM\_BOOL *transmit*, UEM\_WORD *mode*, UEM\_WORD *modecode*, UEM\_OBJHANDLE \* *resp*)

Выяснение ответа на команду управления.

Данная функция позволяет выяснить, какой ответный сегмент установлен функцией **uem\_rt\_install\_response\_MODE()** в качестве ответа на указанную команду. Аргументы полностью аналогичны параметрам **uem\_rt\_install\_response\_MODE()**.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *rt* | Дескриптор виртуального ОУ. |
| in | *transmit* | Признак Передачи(1)/Приема(0) в командном слове. |
| in | *mode* | Код режима управления. Допускаются значения 0 (**UEM\_MODE\_0**) и 31 (**UEM\_MODE\_31**). |
| in | *modecode* | Код команды управления. |
| out | *resp* | Дескриптор ответного сегмента в ОЗУ ОУ. Если ответ на указанную команду не установлен, в качестве значения в этом аргументе будет возвращен NULL. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_response\_destroy (UEM\_OBJHANDLE *resp*)

Уничтожение объекта ответного сегмента в ОЗУ ОУ.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *resp* | Дескриптор ответного сегмента в ОЗУ ОУ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

## Функции МШ

Описания функций монитора шины.

### Определения типов

* typedef void(\* **uem\_bm\_handler**) (**UEM\_DEVHANDLE** bm, void \*userdata)

*Обработчик события МШ.*

### Функции

* ViStatus **uem\_bm\_receive** (**UEM\_DEVHANDLE** bm, **UEM\_BM\_MESSAGE** \*\*message\_data)

*Считывание очередного сообщения, принятого монитором шины.*

* ViStatus **uem\_bm\_queue\_count** (**UEM\_DEVHANDLE** bm, **UEM\_DWORD** \*count)

*Запрос размера очереди сообщений, принятых монитором.*

* ViStatus **uem\_bm\_install\_handler** (**UEM\_DEVHANDLE** bm, **uem\_bm\_handler** handler, void \*userdata)

*Установка обработчика события МШ.*

### Подробное описание

Описания функций монитора шины.

Для использования этих функций необходимо получить дескриптор виртуального МШ при помощи функции **uem\_bm\_init()**.

Запуск МШ в работу выполняется функцией **uem\_start()**. При этом МШ начинает принимать с МКПД и записывать в свою внутреннюю очередь сообщения МКПД. Приложение считывает эти сообщения из очереди при помощи функции **uem\_bm\_receive()**. Остановка МШ выполняется функцией **uem\_stop()**. При этом прием данных МКПД прекращается. Следует иметь в виду, что после остановке МШ в его внутренней очереди могут оставаться сообщения, еще не считанные приложением, которые следует дочитать при помощи **uem\_bm\_receive()**.

### Типы

#### typedef void( \* uem\_bm\_handler) (UEM\_DEVHANDLE bm, void \*userdata)

Обработчик события МШ.

Тип указателя на функцию-обработчика события, вызываемую монитором шины при приходе нового сообщения (сообщений) МКПД.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bm* | Дескриптор виртуального МШ. |
| in | *userdata* | Указатель на произвольные данные приложения, установленный при установке обработчика. |

### Функции

#### ViStatus uem\_bm\_receive (UEM\_DEVHANDLE *bm*, UEM\_BM\_MESSAGE \*\* *message\_data*)

Считывание очередного сообщения, принятого монитором шины.

Функция извлекает из буфера МШ очередное сообщение. Ситуация, когда в буфере МШ нет принятых и несчитанных сообщений, индицируется кодом завершения **UEM\_WARN\_NO\_NEXT\_MESSAGE**.

Внутренняя очередь принятых сообщений может вместить до 210000 сообщений; при максимальном темпе передачи сообщений МКПД очереди хватает на 5 сек., при более низком темпе - на больший интервал времени. Приложение должно считывать все сообщения из очереди не реже 1 раза в 5 сек, рекомендуемый темп - 20 раз в секунду.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bm* | Дескриптор виртуального МШ. |
| out | *message\_data* | Через эту переменную передается указатель на ячейку памяти, в которую МШ записывает указатель на извлеченное сообщение. В случае отсутствия очередного сообщения или ошибки в работе функция записывает в эту переменную значение NULL. Память для извлеченного сообщения выделяется при помощи malloc(), и приложение ответственно за последующее освобождение этой памяти при помощи free(). |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_bm\_queue\_count (UEM\_DEVHANDLE *bm*, UEM\_DWORD \* *count*)

Запрос размера очереди сообщений, принятых монитором.

Функция сообщает приложению количество сообщений МКПД, принятых МШ и ожидающих считывания.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bm* | Дескриптор виртуального МШ. |
| out | *count* | Число сообщений в очереди. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_bm\_install\_handler (UEM\_DEVHANDLE *bm*, uem\_bm\_handler *handler*, void \* *userdata*)

Установка обработчика события МШ.

Указанная функция будет вызываться МШ при получении очередного сообщения или группы сообщений МКПД.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bm* | Дескриптор виртуального МШ. |
| in | *handler* | Функция-обработчик события. Этот аргумент также может быть NULL, что означает отмену функции-обработчика события. |
| in | *userdata* | Указатель на произвольные данные приложения. Этот указатель будет передаваться в функцию-обработчик события handler при каждом вызове. Если приложению такой указатель не требуется, следует указать NULL. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

## Запуск и остановка

Функции запуска, остановки и проверки активности виртуальных устройств.

### Функции

* ViStatus **uem\_start** (**UEM\_DEVHANDLE** anydev)

*Запуск любого виртуального устройства в составе УЭМ.*

* ViStatus **uem\_stop** (**UEM\_DEVHANDLE** anydev)

*Остановка любого виртуального устройства в составе УЭМ.*

* ViStatus **uem\_is\_running** (**UEM\_DEVHANDLE** anydev, **UEM\_BOOL** \*running)

*Проверка активности виртуального устройства.*

### Запуск виртуального КШ с дополнительными параметрами

* enum **UEM\_BC\_STATE\_EX** { **UEM\_BC\_STOPPED**, **UEM\_BC\_WAITING**, **UEM\_BC\_RUNNING**} *Расширенное состояние КШ.*
* ViStatus **uem\_bc\_start** (**UEM\_DEVHANDLE** bc, **UEM\_DWORD** flags)

*Запуск виртуального КШ в составе УЭМ с дополнительными параметрами.*

* ViStatus **uem\_bc\_state\_ex** (**UEM\_DEVHANDLE** bc, **UEM\_BC\_STATE\_EX** \*state)

*Запрос расширенного состояния КШ.*

* #define **UEM\_BC\_START\_NOW**  0

*Нормальный (немедленный) старт.*

* #define **UEM\_BC\_START\_WAITING**  1

*Переход в режим ожидания, старт по внешнему сигналу или команде.*

* #define **UEM\_BC\_START\_DEFAULT**  (**UEM\_BC\_START\_NOW**)

*Стандартный способ старта.*

### Остановка виртуального КШ с дополнительными параметрами

* ViStatus **uem\_bc\_stop** (**UEM\_DEVHANDLE** bc, **UEM\_DWORD** flags)

*Остановка виртуального КШ в составе УЭМ с дополнительными параметрами.*

* #define **UEM\_BC\_STOP\_NOW**  0

*Нормальная (немедленная) остановка.*

* #define **UEM\_BC\_STOP\_ON\_FRAME**  1

*Остановка по завершению текущего кадра.*

* #define **UEM\_BC\_STOP\_DEFAULT**  (**UEM\_BC\_STOP\_NOW**)

*Стандартный способ остановки.*

### Подробное описание

Функции запуска, остановки и проверки активности виртуальных устройств.

Данный раздел содержит описание функций запуска, остановки и проверки активности виртуальных устройств (КШ/ОУ/МШ) в составе УЭМ.

Это универсальные функции, не зависящие от типа виртуального устройства. Для виртуального КШ, при необходимости запуска и остановки с дополнительными параметрами, имеются также специальные функции для КШ.

Перед запуском виртуальное устройство должно быть сконфигурировано при помощи функций соответствующего раздела.

### Макросы

#### #define UEM\_BC\_START\_NOW  0

Нормальный (немедленный) старт.

#### #define UEM\_BC\_START\_WAITING  1

Переход в режим ожидания, старт по внешнему сигналу или команде.

В режиме ожидания старт выполняется в следующих случаях:

При получении внешнего сигнала синхронизации **sync\_in\_2** [**2**, **3**].

При срабатывании внутреннего имитатора внешнего сигнала синхронизации **sync\_in\_2** (см. **UEM\_SYNC\_IN\_2\_INTGEN**, **UEM\_IST2**, а также и **UEM\_SYNC\_IN\_2\_SET**).

При получении виртуальным ОУ в составе УЭМ команды управления "Принять управления интерфейсом" (см. **uem\_response\_create()**, **UEM\_RTDES\_DBCA\_BCSTART**).

Для выполнения такого запуска хотя бы один из указанных механизмов должен быть разрешен и сконфигурирован. В противном случае КШ остается в состоянии ожидания до выполнения действия **uem\_stop()**.

Для определения состояния КШ, запущенного в режиме ожидания, можно использовать функцию **uem\_bc\_state\_ex()**.

#### #define UEM\_BC\_START\_DEFAULT  (UEM\_BC\_START\_NOW)

Стандартный способ старта.

#### #define UEM\_BC\_STOP\_NOW  0

Нормальная (немедленная) остановка.

#### #define UEM\_BC\_STOP\_ON\_FRAME  1

Остановка по завершению текущего кадра.

#### #define UEM\_BC\_STOP\_DEFAULT  (UEM\_BC\_STOP\_NOW)

Стандартный способ остановки.

### Перечисления

#### enum UEM\_BC\_STATE\_EX

Расширенное состояние КШ.

**Элементы перечислений:**

***UEM\_BC\_STOPPED*** Остановлен.

***UEM\_BC\_WAITING*** Находится в режиме ожидания.

***UEM\_BC\_RUNNING*** Работает.

### Функции

#### ViStatus uem\_start (UEM\_DEVHANDLE *anydev*)

Запуск любого виртуального устройства в составе УЭМ.

См. также **uem\_bc\_start()** для виртуального КШ.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *anydev* | Дескриптор виртуального устройства (КШ, ОУ или МШ). |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_stop (UEM\_DEVHANDLE *anydev*)

Остановка любого виртуального устройства в составе УЭМ.

См. также **uem\_bc\_stop()** для виртуального КШ.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *anydev* | Дескриптор виртуального устройства (КШ, ОУ или МШ). |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_is\_running (UEM\_DEVHANDLE *anydev*, UEM\_BOOL \* *running*)

Проверка активности виртуального устройства.

Функция проверяет, работает ли виртуальное устройство в данный момент. Необходимость такой проверки связана с тем, что КШ и ОУ могут останавливаться самостоятельно в зависимости от заданной конфигурации и происходящих событий.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *anydev* | Дескриптор виртуального устройства. |
| out | *running* | В эту переменную записывается результат проверки: 1 - устройство работает, 0 - устройство остановлено. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_bc\_start (UEM\_DEVHANDLE *bc*, UEM\_DWORD *flags*)

Запуск виртуального КШ в составе УЭМ с дополнительными параметрами.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bc* | Дескриптор виртуального КШ. |
| in | *flags* | Дополнительные параметры. Возможные значения **UEM\_BC\_START\_NOW**, **UEM\_BC\_START\_WAITING**. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_bc\_state\_ex (UEM\_DEVHANDLE *bc*, UEM\_BC\_STATE\_EX \* *state*)

Запрос расширенного состояния КШ.

Данная функция может использоваться в дополнение или вместо функции **uem\_is\_running()** для КШ, запущенного в режиме ожидания (**UEM\_BC\_START\_WAITING**).

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bc* | Дескриптор виртуального КШ. |
| out | *state* | Расширенное состояние КШ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_bc\_stop (UEM\_DEVHANDLE *bc*, UEM\_DWORD *flags*)

Остановка виртуального КШ в составе УЭМ с дополнительными параметрами.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *bc* | Дескриптор виртуального КШ. |
| in | *flags* | Дополнительные параметры. Возможные значения **UEM\_BC\_STOP\_NOW**, **UEM\_BC\_STOP\_ON\_FRAME**. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

При остановке по завершению текущего кадра (**flags** == **UEM\_BC\_STOP\_ON\_FRAME**) функция выполняется как неблокирующая. Остановка КШ произойдет по завершению текущего кадра, что зависит от состава и параметров кадра. Для определения фактического состояния КШ можно использовать функции **uem\_is\_running()** и **uem\_bc\_state\_ex()**. Для немедленной остановки КШ, не дожидаясь завершения текущего кадра, допускается повторно вызвать данную функцию с параметром **flags** == **UEM\_BC\_STOP\_NOW**.

## Служебные функции

Описания служебных функций, стандартных для драйвера инструмента.

### Макросы

* #define **UEM\_LIB\_REV**  0x0100

*Номер версии библиотеки.*

### Функции

* ViStatus **uem\_error\_message** (**UEM\_DEVHANDLE** uem, ViStatus status, ViChar msg[])

*Запрос сообщения об ошибке.*

* ViStatus **uem\_error\_query** (ViSession uem, ViInt32 \*status, ViChar msg[])

*Запрос последней ошибки.*

* ViStatus **uem\_reset** (**UEM\_DEVHANDLE** anydev)

*Сброс УЭМ или любого виртуального устройства в составе УЭМ.*

* ViStatus **uem\_revision\_query** (**UEM\_DEVHANDLE** uem, ViChar dv[], ViChar iv[])

*Запрос версии.*

* ViStatus **uem\_self\_test** (**UEM\_DEVHANDLE** uem, ViInt16 \*res, ViChar msg[])

*Самоконтроль.*

### Подробное описание

Описания служебных функций, стандартных для драйвера инструмента.

### Макросы

#### #define UEM\_LIB\_REV  0x0100

Номер версии библиотеки.

В формате: ((старшая часть номера) << 8) | (младшая часть номера).

### Функции

#### ViStatus uem\_error\_message (UEM\_DEVHANDLE *uem*, ViStatus *status*, ViChar *msg*[])

Запрос сообщения об ошибке.

Данная функция получает и интерпретирует код завершения, возвращенный какой-либо функцией драйвера, и возвращает строку сообщения об ошибке. Если код завершения сгенерирован не данным драйвером, функция обращается для получения строки сообщения к аналогичной функции нижележащего драйвера unmuem.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *uem* | Дескриптор УЭМ.  **Примечание** 1: В данной функции в этом аргументе допускается передавать VI\_NULL.  **Примечание** 2: В данной функции в этом аргументе допускается вместо дескриптора УЭМ передавать дескриптор любого виртуального устройства в составе УЭМ. Функция выполняет такой вызов, как если бы был передан дескриптор УЭМ. |
| in | *status* | Код завершения, возвращенный какой-либо функцией драйвера. |
| out | *msg* | Сообщение об ошибке. В данной строке возвращается сообщение, соответствующее переданному коду состояния.  **Примечание:** Строка должна содержать 256 элементов ViChar. |

##### Возвращает:

Код завершения. Возможные значения:

* VI\_SUCCESS - код ошибки интерпретирован успешно.
* VI\_WARN\_UNKNOWN\_STATUS - код ошибки неизвестен.

#### ViStatus uem\_error\_query (ViSession *uem*, ViInt32 \* *status*, ViChar *msg*[])

Запрос последней ошибки.

Возвращает код ошибки последней операции и соответствующее текстовое сообщение. Функция не реализована.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *uem* | Дескриптор УЭМ. |
| out | *status* | В данной переменной возвращается код ошибки, считанный из очереди ошибок инструмента. |
| out | *msg* | В данной строке возвращается сообщение об ошибке, соответствующее коду ошибки.  **Примечание:** Строка должна содержать 256 элементов ViChar. |

##### Возвращает:

Код завершения. Всегда возвращается VI\_ERROR\_NIMPL\_OPER.

#### ViStatus uem\_reset (UEM\_DEVHANDLE *anydev*)

Сброс УЭМ или любого виртуального устройства в составе УЭМ.

Для УЭМ в целом или для указанного виртуального устройства выполняется аппаратный сброс. Кроме этого, для виртуальных КШ и ОУ уничтожаются все объекты в ОЗУ. При указании дескриптора УЭМ все виртуальные устройства в составе этого УЭМ также будут сброшены. Если виртуальное устройство было запущено в работу, оно будет остановлено.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *anydev* | Дескриптор УЭМ или виртуального устройства в составе УЭМ. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_revision\_query (UEM\_DEVHANDLE *uem*, ViChar *dv*[], ViChar *iv*[])

Запрос версии.

Данная функция возвращает версию драйвера и микропрограммы инструмента.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *uem* | Дескриптор УЭМ.  **Примечание:** В данной функции в этом аргументе допускается передавать VI\_NULL. |
| out | *dv* | Версия драйвера.  **Примечание:** Строка должна содержать 256 элементов ViChar. |
| out | *iv* | Версия инструмента.  **Примечание:** Строка должна содержать 256 элементов ViChar.  **Примечание:** Если в функцию передан сеанс NULL, будет возвращена строка "n/a". |

##### Возвращает:

Код завершения. См. **Коды завершения**.

#### ViStatus uem\_self\_test (UEM\_DEVHANDLE *uem*, ViInt16 \* *res*, ViChar *msg*[])

Самоконтроль.

Данная функция производит самоконтроль инструмента и возвращает его результат.

Во время проведения самоконтроля все виртуальные устройства должны быть закрыты.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *uem* | Дескриптор УЭМ. |
| out | *res* | Результат самоконтроля. Нулевое значение означает успешное прохождение самоконтроля. Прочие коды свидетельствуют об ошибке. |
| out | *msg* | Текстовое описание результата самоконтроля. Строка должна содержать 256 элементов ViChar. |

##### Возвращает:

Код завершения. См. **Коды завершения**.

Возможные результаты (res) и соответствующие сообщения (msg):

* 0 - Тест самоконтроля успешно выполнен.
* -1 - Отказ при выполнении теста самоконтроля.
* -2 - Тест самоконтроля выявил ошибки.

## Формирование и разбор командных и ответных слов

Функции формирования и разбора командных и ответных слов.

### Функции

* **UEM\_WORD** **uem\_command\_word** (**UEM\_WORD** rt, **UEM\_BOOL** tx, **UEM\_WORD** sa, **UEM\_WORD** ndatawords)

*Формирование командного слова.*

* void **uem\_command\_word\_parse** (**UEM\_WORD** cw, **UEM\_WORD** \*rt, **UEM\_BOOL** \*tx, **UEM\_WORD** \*sa, **UEM\_WORD** \*ndatawords)

*Разбор командного слова.*

* **UEM\_WORD** **uem\_status\_word** (**UEM\_WORD** rt, **UEM\_WORD** status\_bits)

*Формирование ответного слова.*

* void **uem\_status\_word\_parse** (**UEM\_WORD** sw, **UEM\_WORD** \*rt, **UEM\_WORD** \*status\_bits)

*Разбор ответного слова.*

### Вспомогательные константы для полей командных слов

В соответствии с [**1**].

* #define **UEM\_RTADDR\_MIN**  0

*Минимальное значение адреса ОУ.*

* #define **UEM\_RTADDR\_MАХ**  30

*Максимальное значение адреса ОУ.*

* #define **UEM\_RTADDR\_BRCST**  31

*Адрес для групповых сообщений.*

* #define **UEM\_RTADDR\_MAX\_EXT**  31

*Максимальное значение адреса ОУ в сетях с запретом групповых сообщений.*

* #define **UEM\_RT\_RX**  0

*Признак передача/прием: прием.*

* #define **UEM\_RT\_TX**  1

*Признак передача/прием: передача.*

* #define **UEM\_SADDR\_MIN**  1

*Минимальное значение подадреса.*

* #define **UEM\_SADDR\_MAX**  30

*Максимальное значение подадреса.*

* #define **UEM\_SADDR\_CONV\_LOOPBACK**  30

*Традиционный подадрес для тестовой петли.*

* #define **UEM\_SADDR\_MIN\_EXT**  0

*Минимальное значение подадреса в сетях с запретом команд управления.*

* #define **UEM\_SADDR\_MAX\_EXT**  31

*Максимальное значение подадреса в сетях с запретом команд управления.*

* #define **UEM\_MODE\_0**  0

*Код режима 0.*

* #define **UEM\_MODE\_31**  31

*Код режима 31.*

### Коды команд управления

В соответствии с [**1**].

* #define **UEM\_MCODE\_ADBC**  0

*Принять управление интерфейсом.*

* #define **UEM\_MCODE\_SYNCHRO**  1

*Синхронизация.*

* #define **UEM\_MCODE\_TXSTATUS**  2

*Передать ОС.*

* #define **UEM\_MCODE\_BSELFTEST**  3

*Начать самоконтроль.*

* #define **UEM\_MCODE\_BTMT**  4

*Блокировать передатчик.*

* #define **UEM\_MCODE\_UBTMT**  5

*Разблокировать передатчик.*

* #define **UEM\_MCODE\_BRTF**  6

*Блокировать признак неисправности ОУ.*

* #define **UEM\_MCODE\_UBRTF**  7

*Разблокировать признак неисправности ОУ.*

* #define **UEM\_MCODE\_RESETRT**  8

*Установить ОУ в исходное состояние.*

* #define **UEM\_MCODE\_TXVECT**  16

*Передать векторное слово.*

* #define **UEM\_MCODE\_SYNCHRO\_D**  17

*Синхронизация (с СД).*

* #define **UEM\_MCODE\_TXLCMD**  18

*Передать последнюю команду.*

* #define **UEM\_MCODE\_TXBIT**  19

*Передать слово ВСК ОУ.*

* #define **UEM\_MCODE\_BTMT\_I**  20

*Блокировать i-й передатчик.*

* #define **UEM\_MCODE\_UBTMT\_I**  21

*Разблокировать i-й передатчик.*

### Признаки ответного слова

В соответствии с [**1**].

* #define **UEM\_RTFAIL**  0x0001

*НОУ - Неисправность оконечного устройства.*

* #define **UEM\_DBCA**  0x0002

*ПУИ - Принято управление интерфейсом.*

* #define **UEM\_ABFAIL**  0x0004

*НА - Неисправность абонента.*

* #define **UEM\_ABBUSY**  0x0008

*АЗ - Абонент занят.*

* #define **UEM\_BCCA**  0x0010

*ПГК - Принята групповая команда.*

* #define **UEM\_RSV14**  0x0020

*Резервный бит 14.*

* #define **UEM\_RSV13**  0x0040

*Резервный бит 13.*

* #define **UEM\_RSV12**  0x0080

*Резервный бит 12.*

* #define **UEM\_SERVRQ**  0x0100

*ЗО - Запрос обслуживания.*

* #define **UEM\_ZERO**  0x0200

*Передача ОС.*

* #define **UEM\_MSGERR**  0x0400

*ОШС - Ошибка в сообщении.*

### Подробное описание

Функции формирования и разбора командных и ответных слов.

Функции данного раздела служат для сборки командных и ответных слов из полей и разбора на поля.

Для этих целей также можно использовать структуры данных из unmuem\_struct.h [**5**, **6**]

### Макросы

#### #define UEM\_RTADDR\_MIN  0

Минимальное значение адреса ОУ.

#### #define UEM\_RTADDR\_MАХ  30

Максимальное значение адреса ОУ.

#### #define UEM\_RTADDR\_BRCST  31

Адрес для групповых сообщений.

#### #define UEM\_RTADDR\_MAX\_EXT  31

Максимальное значение адреса ОУ в сетях с запретом групповых сообщений.

#### #define UEM\_RT\_RX  0

Признак передача/прием: прием.

#### #define UEM\_RT\_TX  1

Признак передача/прием: передача.

#### #define UEM\_SADDR\_MIN  1

Минимальное значение подадреса.

#### #define UEM\_SADDR\_MAX  30

Максимальное значение подадреса.

#### #define UEM\_SADDR\_CONV\_LOOPBACK  30

Традиционный подадрес для тестовой петли.

#### #define UEM\_SADDR\_MIN\_EXT  0

Минимальное значение подадреса в сетях с запретом команд управления.

#### #define UEM\_SADDR\_MAX\_EXT  31

Максимальное значение подадреса в сетях с запретом команд управления.

#### #define UEM\_MODE\_0  0

Код режима 0.

#### #define UEM\_MODE\_31  31

Код режима 31.

#### #define UEM\_MCODE\_ADBC  0

Принять управление интерфейсом.

#### #define UEM\_MCODE\_SYNCHRO  1

Синхронизация.

#### #define UEM\_MCODE\_TXSTATUS  2

Передать ОС.

#### #define UEM\_MCODE\_BSELFTEST  3

Начать самоконтроль.

#### #define UEM\_MCODE\_BTMT  4

Блокировать передатчик.

#### #define UEM\_MCODE\_UBTMT  5

Разблокировать передатчик.

#### #define UEM\_MCODE\_BRTF  6

Блокировать признак неисправности ОУ.

#### #define UEM\_MCODE\_UBRTF  7

Разблокировать признак неисправности ОУ.

#### #define UEM\_MCODE\_RESETRT  8

Установить ОУ в исходное состояние.

#### #define UEM\_MCODE\_TXVECT  16

Передать векторное слово.

#### #define UEM\_MCODE\_SYNCHRO\_D  17

Синхронизация (с СД).

#### #define UEM\_MCODE\_TXLCMD  18

Передать последнюю команду.

#### #define UEM\_MCODE\_TXBIT  19

Передать слово ВСК ОУ.

#### #define UEM\_MCODE\_BTMT\_I  20

Блокировать i-й передатчик.

#### #define UEM\_MCODE\_UBTMT\_I  21

Разблокировать i-й передатчик.

#### #define UEM\_RTFAIL  0x0001

НОУ - Неисправность оконечного устройства.

#### #define UEM\_DBCA  0x0002

ПУИ - Принято управление интерфейсом.

#### #define UEM\_ABFAIL  0x0004

НА - Неисправность абонента.

#### #define UEM\_ABBUSY  0x0008

АЗ - Абонент занят.

#### #define UEM\_BCCA  0x0010

ПГК - Принята групповая команда.

#### #define UEM\_RSV14  0x0020

Резервный бит 14.

#### #define UEM\_RSV13  0x0040

Резервный бит 13.

#### #define UEM\_RSV12  0x0080

Резервный бит 12.

#### #define UEM\_SERVRQ  0x0100

ЗО - Запрос обслуживания.

#### #define UEM\_ZERO  0x0200

Передача ОС.

#### #define UEM\_MSGERR  0x0400

ОШС - Ошибка в сообщении.

### Функции

#### UEM\_WORD uem\_command\_word (UEM\_WORD *rt*, UEM\_BOOL *tx*, UEM\_WORD *sa*, UEM\_WORD *ndatawords*)

Формирование командного слова

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *rt* | Адрес ОУ. |
| in | *tx* | Разряд Передача(1)/Прием(0). |
| in | *sa* | Подадрес. |
| in | *ndatawords* | Число слов данных. |

##### Возвращает:

Командное слово.

Для команд управления в **sa** следует указывать режим управления, а в **ndatawords** - код команды управления.

##### См. также:

**Вспомогательные константы**, **Коды команд управления**.

#### void uem\_command\_word\_parse (UEM\_WORD *cw*, UEM\_WORD \* *rt*, UEM\_BOOL \* *tx*, UEM\_WORD \* *sa*, UEM\_WORD \* *ndatawords*)

Разбор командного слова

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *cw* | Командное слово. |
| out | *rt* | Адрес ОУ. |
| out | *tx* | Разряд Передача(1)/Прием(0). |
| out | *sa* | Подадрес. |
| out | *ndatawords* | Число слов данных. |

##### Возвращает:

Функция не возвращает значения.

#### UEM\_WORD uem\_status\_word (UEM\_WORD *rt*, UEM\_WORD *status\_bits*)

Формирование ответного слова.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *rt* | Адрес ОУ. |
| in | *status\_bits* | Признаки ответного слова. Объединение (по |) констант **Признаков ответного слова**. |

##### Возвращает:

Ответное слово.

#### void uem\_status\_word\_parse (UEM\_WORD *sw*, UEM\_WORD \* *rt*, UEM\_WORD \* *status\_bits*)

Разбор ответного слова.

##### Аргументы:

|  |  |  |
| --- | --- | --- |
| in | *sw* | Ответное слово. |
| out | *rt* | Адрес ОУ. |
| out | *status\_bits* | Признаки ответного слова. Объединение (по |) констант **Признаков ответного слова**. |

##### Возвращает:

Функция не возвращает значения.

# Структуры данных

## Структура UEM\_BM\_MESSAGE

Разобранное сообщение МШ.

#include <uem.h>

### Поля данных

* **UEM\_CMD\_SEG** **cs**

*Командный сегмент, включая формат сообщения и селектор шины.*

* **UEM\_BOOL** **rs1\_pr**

*Признак наличия ответного сегмента 1.*

* **UEM\_BOOL** **rs2\_pr**

*Признак наличия ответного сегмента 2.*

* **UEM\_RESP\_SEG** **rs1**

*Ответный сегмент 1.*

* **UEM\_RESP\_SEG** **rs2**

*Ответный сегмент 2.*

* **UEM\_SEGMENT\_DESCR** **msg\_d**

*Описатель сообщения в целом.*

* **UEM\_SEGMENT\_DESCR** **cs\_d**

*Описатель командного сегмента.*

* **UEM\_SEGMENT\_DESCR** **rs1\_d**

*Описатель ответного сегмента 1.*

* **UEM\_SEGMENT\_DESCR** **rs2\_d**

*Описатель ответного сегмента 2.*

* **UEM\_ERROR\_FLAGS** **errors**

*Строка бит - признаков ошибок распознавания сообщения.*

* **UEM\_BOOL** **p\_pr**

*Признак наличия предыдущего сообщения.*

* **UEM\_BOOL** **overlay**

*Признак наложения на предыдущее сообщение.*

* **UEM\_TIME\_TAG\_LIN** **gap**

*Пауза между предыдущим и данным сообщением.*

* **UEM\_DWORD** **lostp**

*Количество пропущенных предыдущих сообщений.*

* **UEM\_RAW\_BM\_MESSAGE** **raw**

*Неразобранное сообщение аппаратного формата.*

### Подробное описание

Разобранное сообщение МШ, представление в ОЗУ управляющей ПЭВМ, генерируется программным обеспечением виртуального МШ и содержит

* идентифицированные части (сегменты) сообщения,
* описатели сегментов сообщения, включая
* - время начала и конца,
* - признаки ошибок распознавания,
* - расположение сегмента в неразобранном сообщении в аппаратном формате,
* аналогичный описатель для сообщения в целом,
* признаки ошибок распознавания,
* пауза между предыдущим и данным сообщением,
* количество пропущенных предыдущих сообщений,
* неразобранное сообщение МШ в аппаратном формате.

**Примечание:** даже при сброшенных **rs1\_pr** и **rs2\_pr** признаки ошибки rs1\_d.errors и rs2\_d.errors могут содержать признак ошибки "Отсутствие ответа" (если формат сообщения предполагает ответ).

### Поля

#### UEM\_CMD\_SEG cs

Командный сегмент, включая формат сообщения и селектор шины.

#### UEM\_BOOL rs1\_pr

Признак наличия ответного сегмента 1.

#### UEM\_BOOL rs2\_pr

Признак наличия ответного сегмента 2.

#### UEM\_RESP\_SEG rs1

Ответный сегмент 1.

Заполнено только если **rs1\_p** выставлен в 1.

#### UEM\_RESP\_SEG rs2

Ответный сегмент 2.

Заполнено только если **rs2\_p** выставлен в 1.

#### UEM\_SEGMENT\_DESCR msg\_d

Описатель сообщения в целом.

#### UEM\_SEGMENT\_DESCR cs\_d

Описатель командного сегмента.

#### UEM\_SEGMENT\_DESCR rs1\_d

Описатель ответного сегмента 1.

Поле errors заполнено всегда, остальные поля - только если **rs1\_p** выставлен в 1.

#### UEM\_SEGMENT\_DESCR rs2\_d

Описатель ответного сегмента 2.

Поле **errors** заполнено всегда, остальные поля - только если **rs2\_p** выставлен в 1.

#### UEM\_ERROR\_FLAGS errors

Строка бит - признаков ошибок распознавания сообщения.

#### UEM\_BOOL p\_pr

Признак наличия предыдущего сообщения.

#### UEM\_BOOL overlay

Признак наложения на предыдущее сообщение.

#### UEM\_TIME\_TAG\_LIN gap

Пауза между предыдущим и данным сообщением.

Заполняется только при p\_pr == 1. При overlay == 1 это сдвиг начала данного сообщения по отношению к началу предыдущего. При overlay == 0 это пауза между концом предыдущего и началом данного сообщения, измеренная по ГОСТ [**1**], т.е. от последнего перепада предыдущего сообщения до первого перепада данного сообщения (+2 мкс по сравнению со "временем тишины", получаемой вычитанием моментов времени исчезновения и появления сигнала).

#### UEM\_DWORD lostp

Количество пропущенных предыдущих сообщений.

Пропуск возможен при переполнении очереди сообщений или недостатка ОЗУ управляющей ПЭВМ. Для исключения пропуска приложение должно быстрее выбирать сообщения из очереди (см. **uem\_bm\_receive()**).

#### UEM\_RAW\_BM\_MESSAGE raw

Неразобранное сообщение аппаратного формата.

#### Объявления и описания членов структуры находятся в файле:

uem.h

## Структура UEM\_CMD\_SEG

Образ командного сегмента.

#include <uem.h>

### Поля данных

* **UEM\_CHANNEL** **ch**

*Шина.*

* **UEM\_FORMAT** **format**

*Формат сообщения.*

* **UEM\_WORD** **command1**

*Командное слово.*

* **UEM\_WORD** **command2**

*Второе командное слово в сообщениях форматов 3 и 8.*

* **UEM\_DATA** **data**

*Слова данных.*

### Подробное описание

Образ командного сегмента.

Представление в ОЗУ управляющей ПЭВМ.

Использование полей **command1** и **command2** определяется форматом сообщения. Наличие, количество и состав слов данных определяется полем **data**.

В неформатных сообщениях (**UEM\_UNF**) поля **command1** и **command2** не используются, сегмент состоит только из слов данных. Если необходимо, чтобы в сегменте были командные слова, следует установить в этих словах тип синхроимпульса командного слова (**UEM\_SYNC\_C**) при помощи функции **uem\_cseg\_sync\_set()**, после создания командного сегмента.

### Поля

#### UEM\_CHANNEL ch

Шина.

#### UEM\_FORMAT format

Формат сообщения.

#### UEM\_WORD command1

Командное слово.

#### UEM\_WORD command2

Второе командное слово в сообщениях форматов 3 и 8.

#### UEM\_DATA data

Слова данных.

#### Объявления и описания членов структуры находятся в файле:

uem.h

## Структура UEM\_DATA

Блок слов данных.

#include <uem.h>

### Поля данных

* **UEM\_WORD** **ndata**

*Число слов данных.*

* **UEM\_WORD** **data** [62]

*Слова данных.*

### Подробное описание

Блок слов данных.

### Поля

#### UEM\_WORD ndata

Число слов данных.

#### UEM\_WORD data[62]

Слова данных.

**Примечание:** Аппаратно УЭМ может формировать командные и ответные сегменты, содержащие до 62 слов данных.

#### Объявления и описания членов структуры находятся в файле:

uem.h

## Структура UEM\_RAW\_BM\_MESSAGE

Принятое сообщение в аппаратном формате.

#include <uem.h>

### Поля данных

* **UEM\_DWORD** **size**

*Размер массива записей трассы.*

* unmuem\_mt\_data\_t **data** [0]

*Массив записей трассы.*

### Подробное описание

Принятое сообщение в аппаратном формате генерируется аппаратурой монитора шины и состоит из последовательности записей трассы unmuem\_mt\_data\_t. Определение этой структуры данных имеется в файле unmuem\_struct.h [**5**, **6**], а описание ее полей - в [**2**, **3**].

### Поля

#### UEM\_DWORD size

Размер массива записей трассы.

#### unmuem\_mt\_data\_t data[0]

Массив записей трассы.

#### Объявления и описания членов структуры находятся в файле:

uem.h

## Структура UEM\_RESP\_SEG

Ответный сегмент.

#include <uem.h>

### Поля данных

* **UEM\_WORD** **status**

*Ответное слово.*

* **UEM\_DATA** **data**

*Слова данных.*

### Подробное описание

Ответный сегмент.

Представление в ОЗУ управляющей ПЭВМ.

### Поля

#### UEM\_WORD status

Ответное слово.

#### UEM\_DATA data

Слова данных.

#### Объявления и описания членов структуры находятся в файле:

uem.h

## Структура UEM\_SEGMENT\_DESCR

Описатель сегмента в мониторе шины.

#include <uem.h>

### Поля данных

* **UEM\_TIME\_TAG** **start\_time**

*Время начала сегмента (время появления сигнала).*

* **UEM\_TIME\_TAG** **end\_time**

*Время конца сегмента (время исчезновения сигнала).*

* **UEM\_ERROR\_FLAGS** **errors**

*Признаки обнаруженных ошибок.*

* ViUInt16 **offset**

*Смещение сегмента в* ***UEM\_RAW\_BM\_MESSAGE****.*

* ViUInt16 **dwoffset**

*Смещение первого слова данных в* ***UEM\_RAW\_BM\_MESSAGE****.*

* ViUInt16 **endoffset**

*Смещение первого слова за концом данного сегмента.*

* ViUInt16 **size**

*Размер сегмента в словах.*

### Подробное описание

Описатель сегмента в мониторе шины.

### Поля

#### UEM\_TIME\_TAG start\_time

Время начала сегмента (время появления сигнала).

Точность измерений ± 0,25 мкс.

#### UEM\_TIME\_TAG end\_time

Время конца сегмента (время исчезновения сигнала).

Точность измерений ± 0,25 мкс.

#### UEM\_ERROR\_FLAGS errors

Признаки обнаруженных ошибок.

#### ViUInt16 offset

Смещение сегмента в **UEM\_RAW\_BM\_MESSAGE**.

#### ViUInt16 dwoffset

Смещение первого слова данных в **UEM\_RAW\_BM\_MESSAGE**.

#### ViUInt16 endoffset

Смещение первого слова за концом данного сегмента.

#### ViUInt16 size

Размер сегмента в словах.

#### Объявления и описания членов структуры находятся в файле:

uem.h

## Объединение UEM\_TIME\_TAG

Формат метки времени.

#include <uem.h>

### Поля данных

struct {

* ViUInt64 **quartas**:22

*Число единицы метки времени, так называемых четвертей ; 0,25 мкс на единицу младшего разряда; диапазон 0 - 3 999 999 (охватывает 1 сек).*

* ViUInt64 **secs**:17

*Число секунд; 1 сек на единицу младшего разряда; диапазон 0 - 86 399 (охватывает 1 сутки).*

* ViUInt64 **days**:9

*Число суток; 1 сутки на единицу младшего разряда; диапазон 0 - 365 (охватывает 366 суток).*

* ViUInt64 **reserved**:16

*Не используется и всегда содержит 0.*

* } **b**

*Этот элемент объединения обеспечивает доступ к элементам метки времени - перечисленным полям.*

* ViUInt64 **i**

*Этот элемент объединения обеспечивает манипулирование значением как единым целым.*

### Подробное описание

Формат метки времени.

### Поля

#### ViUInt64 quartas

Число единиц метки времени, так называемых *четвертей* ; 0,25 мкс на единицу младшего разряда; диапазон 0 - 3 999 999 (охватывает 1 сек).

#### ViUInt64 secs

Число секунд; 1 сек на единицу младшего разряда; диапазон 0 - 86 399 (охватывает 1 сутки).

#### ViUInt64 days

Число суток; 1 сутки на единицу младшего разряда; диапазон 0 - 365 (охватывает 366 суток).

#### ViUInt64 reserved

Не используется и всегда содержит 0.

#### struct { ... } b

Этот элемент объединения обеспечивает доступ к элементам метки времени - перечисленным полям.

#### ViUInt64 i

Этот элемент объединения обеспечивает манипулирование значением как единым целым.

#### Объявления и описания членов объединения находятся в файле:

uem.h

# Файлы

## Файл uem.h

Универсальные электронные модули УЭМ-МК, МВ98.03. Расширенная библиотека функций. Файл заголовков функций.

#include "unmuem.h"

#include "unmuem\_struct.h"

### Структуры данных

* union **UEM\_TIME\_TAG**

*Формат метки времени.*

* struct **UEM\_DATA**

*Блок слов данных.*

* struct **UEM\_CMD\_SEG**

*Образ командного сегмента.*

* struct **UEM\_RESP\_SEG**

*Ответный сегмент.*

* struct **UEM\_RAW\_BM\_MESSAGE**

*Принятое сообщение в аппаратном формате.*

* struct **UEM\_SEGMENT\_DESCR**

*Описатель сегмента в мониторе шины.*

* struct **UEM\_BM\_MESSAGE**

### *Разобранное сообщение МШ.*

### Макросы

* #define **UEM\_WARN\_OFFSET**  (0x3FFC0B00L)

*Начальный номер кодов предупреждений.*

* #define **UEM\_ERROR\_OFFSET**  (\_VI\_ERROR + **UEM\_WARN\_OFFSET**)

*Начальный номер кодов ошибок.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE**  (**UEM\_ERROR\_OFFSET** + 0)

*Недопустимое значение параметра.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_1**  (**UEM\_ERROR\_OFFSET** + 1)

*Недопустимое значение в параметре 1.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_2**  (**UEM\_ERROR\_OFFSET** + 2)

*Недопустимое значение в параметре 2.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_3**  (**UEM\_ERROR\_OFFSET** + 3)

*Недопустимое значение в параметре 3.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_4**  (**UEM\_ERROR\_OFFSET** + 4)

*Недопустимое значение в параметре 4.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_5**  (**UEM\_ERROR\_OFFSET** + 5)

*Недопустимое значение в параметре 5.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_6**  (**UEM\_ERROR\_OFFSET** + 6)

*Недопустимое значение в параметре 6.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_7**  (**UEM\_ERROR\_OFFSET** + 7)

*Недопустимое значение в параметре 7.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_8**  (**UEM\_ERROR\_OFFSET** + 8)

*Недопустимое значение в параметре 8.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_9**  (**UEM\_ERROR\_OFFSET** + 9)

*Недопустимое значение в параметре 9.*

* #define **UEM\_ERROR\_BAD\_PARAM\_VALUE\_10**  (**UEM\_ERROR\_OFFSET** + 10)

*Недопустимое значение в параметре 10.*

* #define **UEM\_ERROR\_INV\_HANDLE**  (**UEM\_ERROR\_OFFSET** + 11)

*Недействительный дескриптор.*

* #define **UEM\_ERROR\_INV\_HANDLE\_TYPE**  (**UEM\_ERROR\_OFFSET** + 12)

*Неподходящий тип дескриптора.*

* #define **UEM\_ERROR\_NO\_FREE\_RAM**  (**UEM\_ERROR\_OFFSET** + 13)

*Недостаточно ОЗУ УЭМ.*

* #define **UEM\_ERROR\_NO\_HOST\_MEM**  (**UEM\_ERROR\_OFFSET** + 14)

*Недостаточно ОЗУ управляющей ПЭВМ.*

* #define **UEM\_ERROR\_NOT\_CONNECTED**  (**UEM\_ERROR\_OFFSET** + 15)

*Нет связи с устройством.*

* #define **UEM\_ERROR\_INPOOL**  (**UEM\_ERROR\_OFFSET** + 16)

*Внутренняя ошибка менеджера памяти.*

* #define **UEM\_ERROR\_BCP\_NINST**  (**UEM\_ERROR\_OFFSET** + 17)

*Не установлена программа КШ.*

* #define **UEM\_ERROR\_FORMAT\_DISABLED**  (**UEM\_ERROR\_OFFSET** + 18)

*Формат сообщения запрещен конфигурацией УЭМ.*

* #define **UEM\_ERROR\_FORMAT\_X\_MCODE**  (**UEM\_ERROR\_OFFSET** + 19)

*Формат сообщения несовместим с командой управления.*

* #define **UEM\_ERROR\_NOT\_APPLICABLE**  (**UEM\_ERROR\_OFFSET** + 20)

*Действие не применимо к объекту.*

* #define **UEM\_ERROR\_ADDRESS\_OUT\_OF\_RANGE**  (**UEM\_ERROR\_OFFSET** + 21)

*Адрес вне допустимого диапазона.*

* #define **UEM\_ERROR\_NUMBER\_OUT\_OF\_RANGE**  (**UEM\_ERROR\_OFFSET** + 23)

*Номер вне допустимого диапазона.*

* #define **UEM\_ERROR\_BAD\_TIMEOUT**  (**UEM\_ERROR\_OFFSET** + 24)

*Недопустимое значение таймаута.*

* #define **UEM\_ERROR\_BAD\_OVERLAY\_SOURCE**  (**UEM\_ERROR\_OFFSET** + 25)

*Недопустимые исходные сегменты для наложения.*

* #define **UEM\_ERROR\_WRONG\_LOCATION**  (**UEM\_ERROR\_OFFSET** + 26)

*Объект расположен не в том устройстве.*

* #define **UEM\_ERROR\_TOO\_MANY\_DATAWORDS**  (**UEM\_ERROR\_OFFSET** + 27)

*Слишком много слов данных.*

* #define **UEM\_ERROR\_MAX\_SIZE\_EXCEED**  (**UEM\_ERROR\_OFFSET** + 28)

*Превышен максимальный размер.*

* #define **UEM\_ERROR\_NO\_FRAME\_APPEND**  (**UEM\_ERROR\_OFFSET** + 29)

*Не добавлен кадр.*

* #define **UEM\_ERROR\_IN\_USE**  (**UEM\_ERROR\_OFFSET** + 30)

*Устройство или объект используются.*

* #define **UEM\_ERROR\_THREAD\_FAULT**  (**UEM\_ERROR\_OFFSET** + 31)

*Ошибки в работе служебной нити.*

* #define **UEM\_ERROR\_BM\_INTERNAL\_BUFFER\_OVERFLOW**  (**UEM\_ERROR\_OFFSET** + 32)

*Переполнение внутреннего буфера МШ.*

* #define **UEM\_ERROR\_INC\_RESP**  (**UEM\_ERROR\_OFFSET** + 33)

*Несовместимый ответный сегмент.*

* #define **UEM\_WARN\_NO\_NEXT\_MESSAGE**  (**UEM\_WARN\_OFFSET** + 0)

*Нет следующего сообщения (в буфере МШ).*

* #define **UEM\_WARN\_JUST\_IN\_STATE**  (**UEM\_WARN\_OFFSET** + 1)

*Устройство уже в нужном состоянии; никаких действий не производится.*

* #define **UEM\_TMTA\_DIS**  UEMi\_MAKE\_PARAMID ( 0, 2, 2)

*Запрет работы передатчика шины А.*

* #define **UEM\_TMTB\_DIS**  UEMi\_MAKE\_PARAMID ( 0, 3, 3)

*Запрет работы передатчика шины Б.*

* #define **UEM\_RCVA\_DIS**  UEMi\_MAKE\_PARAMID ( 0, 4, 4)

*Запрет работы приемника шины А.*

* #define **UEM\_RCVB\_DIS**  UEMi\_MAKE\_PARAMID ( 0, 5, 5)

*Запрет работы приемника шины Б.*

* #define **UEM\_SYNC\_IN\_1\_ENA**  UEMi\_MAKE\_PARAMID ( 0, 8, 8)

*Разрешение входной синхронизации 1.*

* #define **UEM\_SYNC\_IN\_2\_ENA**  UEMi\_MAKE\_PARAMID ( 0, 9, 9)

*Разрешение входной синхронизации 2.*

* #define **UEM\_SYNC\_OUT\_1\_ENA**  UEMi\_MAKE\_PARAMID ( 0, 10, 10)

*Разрешение выходной синхронизации 1.*

* #define **UEM\_SYNC\_OUT\_2\_ENA**  UEMi\_MAKE\_PARAMID ( 0, 11, 11)

*Разрешение выходной синхронизации 2.*

* #define **UEM\_BRCST\_DIS**  UEMi\_MAKE\_PARAMID ( 0, 12, 12)

*Запрет групповых сообщений.*

* #define **UEM\_SYNC\_IN\_1\_INTGEN**  UEMi\_MAKE\_PARAMID( 0, 13, 13)

*Разрешение внутренней эмуляции сигнала входной синхронизации 1.*

* #define **UEM\_ERR\_INJ\_DIS**  UEMi\_MAKE\_PARAMID ( 0, 14, 14)

*Запрет внесения ошибок кодирования в передаваемую в МКПД информацию для КШ и ОУ.*

* #define **UEM\_SYNC\_IN\_2\_INTGEN**  UEMi\_MAKE\_PARAMID( 0, 29, 29)

*Разрешение внутренней эмуляции сигнала входной синхронизации 2.*

* #define **UEM\_TMT\_RES**  UEMi\_MAKE\_PARAMID ( 5, 4, 4)

*Сброс настроек передатчиков (только запись).*

* #define **UEM\_SYNC\_IN\_1\_SET**  UEMi\_MAKE\_PARAMID ( 5, 10, 10)

*Программная генерация сигнала входной синхронизации 1 (только запись).*

* #define **UEM\_SYNC\_IN\_2\_SET**  UEMi\_MAKE\_PARAMID ( 5, 11, 11)

*Программная генерация сигнала входной синхронизации 2 (только запись).*

* #define **UEM\_SYNC\_OUT\_1\_SET**  UEMi\_MAKE\_PARAMID ( 5, 12, 12)

*Программная генерация сигнала выходной синхронизации 1 (только запись).*

* #define **UEM\_SYNC\_OUT\_2\_SET**  UEMi\_MAKE\_PARAMID ( 5, 13, 13)

*Программная генерация сигнала выходной синхронизации 2 (только запись).*

* #define **UEM\_DB\_ACT**  UEMi\_MAKE\_PARAMID ( 5, 31, 31)

*Обнаружена передача данных по шине (только чтение).*

* #define **UEM\_TXA\_RFT**  UEMi\_MAKE\_PARAMID ( 6, 31, 16)

*Управление длительностью фронта и среза при передаче в шину А.*

* #define **UEM\_TXA\_VPP**  UEMi\_MAKE\_PARAMID ( 6, 15, 0)

*Управление размахом сигнала при передаче в шину А.*

* #define **UEM\_TXB\_RFT**  UEMi\_MAKE\_PARAMID ( 7, 31, 16)

*Управление длительностью фронта и среза при передаче в шину Б.*

* #define **UEM\_TXB\_VPP**  UEMi\_MAKE\_PARAMID ( 7, 15, 0)

*Управление размахом сигнала при передаче в шину Б.*

* #define **UEM\_MC\_DIS**  UEMi\_MAKE\_PARAMID (0x0C, 0, 0)

*Запрет команд управления.*

* #define **UEM\_BTMT\_DIS**  UEMi\_MAKE\_PARAMID (0x0C, 1, 1)

*Запрет блокирования и разблокирования передатчиков ОУ.*

* #define **UEM\_BRTF\_DIS**  UEMi\_MAKE\_PARAMID (0x0C, 2, 2)

*Запрет блокирования и разблокирования признака неисправности ОУ.*

* #define **UEM\_SYNC2\_VRTA**  UEMi\_MAKE\_PARAMID (0x0C, 7, 3)

*Номер ОУ – условие выработки сигнала выходной синхронизации 2.*

* #define **UEM\_SYNC1\_C\_D\_**  UEMi\_MAKE\_PARAMID (0x0C, 8, 8)

*Командное/ответное (1) слово или слово данных (0) – условие выработки сигнала выходной синхронизации 1.*

* #define **UEM\_SYNC1\_ERR**  UEMi\_MAKE\_PARAMID (0x0C, 9, 9)

*Наличие ошибок в слове – условие выработки сигнала выходной синхронизации 1.*

* #define **UEM\_SYNC1\_GAPB**  UEMi\_MAKE\_PARAMID (0x0C, 10, 10)

*Наличие паузы перед словом - условие выработки сигнала выходной синхронизации 1.*

* #define **UEM\_SYNC1\_CH**  UEMi\_MAKE\_PARAMID (0x0C, 12, 11)

*Слово передается по указанной шине - условие выработки сигнала выходной синхронизации 1.*

* #define **UEM\_IST2**  UEMi\_MAKE\_PARAMID (0x0C, 31, 16)

*Период внутренней генерации сигнала входной синхронизации 2.*

* #define **UEM\_IST1**  UEMi\_MAKE\_PARAMID ( 8, 31, 16)

*Период внутренней генерации сигнала входной синхронизации 1.*

* #define **UEM\_BM\_WORD\_PATTERN**  UEMi\_MAKE\_PARAMID (0x0F, 15, 0)

*Шаблон (значение) слова - условие выработки сигнала выходной синхронизации 1.*

* #define **UEM\_BM\_WORD\_MASK**  UEMi\_MAKE\_PARAMID (0x0F, 31, 16)

*Маска побитного сравнения слова с шаблоном - условие выработки сигнала выходной синхронизации 1.*

* #define **UEM\_MIN\_T1\_DEFAULT**  (4\*4)

*Минимальная пауза между командным и ответным сегментами (min t1), значение по умолчанию.*

* #define **UEM\_MIN\_T2\_DEFAULT**  (4\*4)

*Минимальная пауза между сообщениями (min t2), значение по умолчанию.*

* #define **UEM\_RTMO\_DEFAULT**  (14\*4)

*Таймаут ответа ОУ, значение по умолчанию.*

* #define **UEM\_LIB\_REV**  0x0100

*Номер версии библиотеки.*

Граничные значения изменения количества разрядов.

*Эти константы задают граничные значения аргумента* ***error\_pos*** *в функциях* ***uem\_cseg\_error\_set()****,* ***uem\_response\_error\_set()****, когда в аргументе* ***error\_type*** *указано* ***UEM\_ERRT\_WRONG\_BITCOUNT****.*

#define UEM\_BITCOUNT\_CHANGE\_MIN  (-3)

*Минимальное приращение количества разрядов.*

#define UEM\_BITCOUNT\_CHANGE\_MAX  (+3)

*Максимальное приращение количества разрядов.*

Граничные значения позиции ошибки.

*Эти константы задают граничные значения аргумента* ***error\_pos*** *для некоторых значений аргумента* ***error\_type*** *в функциях* ***uem\_cseg\_error\_set()****,* ***uem\_response\_error\_set()****.*

#define UEM\_BIPHASE\_POS\_MIN  4

*Минимальная позиция при* ***error\_type****, равном* ***UEM\_ERRT\_BAD\_BIPHASE\_ZERO****,* ***UEM\_ERRT\_BAD\_BIPHASE\_POS*** *или* ***UEM\_ERRT\_BAD\_BIPHASE\_NEG****.*

#define UEM\_BIPHASE\_POS\_MAX  20

*Максимальная позиция при* ***error\_type****, равном* ***UEM\_ERRT\_BAD\_BIPHASE\_ZERO****,* ***UEM\_ERRT\_BAD\_BIPHASE\_POS*** *или* ***UEM\_ERRT\_BAD\_BIPHASE\_NEG****.*

#define UEM\_SHIFT\_POS\_MIN  0

*Минимальная позиция при* ***error\_type****, равном* ***UEM\_ERRT\_SHIFT\_EDGE****.*

#define UEM\_SHIFT\_POS\_MAX  40

*Максимальная позиция при* ***error\_type****, равном* ***UEM\_ERRT\_SHIFT\_EDGE****.*

Граничные значения величины сдвига.

*Эти константы задают граничные значения для аргумента* ***error\_param*** *в функциях* ***uem\_cseg\_error\_set()****,* ***uem\_response\_error\_set()****, когда аргумент* ***error\_type*** *равен* ***UEM\_ERRT\_SHIFT\_EDGE****. Величина сдвига указывается в единицах по 10 нс. Отрицательные значения обозначают сдвиг влево, положительные - вправо.*

#define UEM\_SHIFT\_LENGTH\_MIN  (-25)

*Максимальная величина сдвига влево.*

#define UEM\_SHIFT\_LENGTH\_MAX  (+25)

*Максимальная величина сдвига вправо.*

Значения аргументов по умолчанию.

*Эти константы задают значения аргументов функций* ***uem\_cseg\_error\_set()****,* ***uem\_response\_error\_set()*** *по умолчанию, соответствуют отсутствию внесения ошибок.*

#define UEM\_ERROR\_TYPE\_DEFAULT  0

*Значение по умолчанию для* ***error\_type****.*

#define UEM\_ERROR\_POS\_DEFAULT  0

*Значение по умолчанию для* ***error\_pos****.*

#define UEM\_ERROR\_PARAM\_DEFAULT  0

*Значение по умолчанию для* ***error\_param****.*

Селектор сеанса.

#define UEM\_SEL\_UNMUEM  1

*Сеанс низкоуровневого драйвера УЭМ unmuem.*

#define UEM\_SEL\_UNMBASE  2

*Сеанc драйвера носителя мезонинов unmbase.*

#define UEM\_SEL\_UNBASE\_INT  3

*"Внутренний" сеанс драйвера носителя мезонинов unmbase.*

Значения параметра UEM\_TMTA\_DIS.

#define UEM\_TMTA\_DISABLED  1

*Передатчик шины А отключен.*

#define UEM\_TMTA\_ENABLED  0

*Передатчик шины А включен.*

#define UEM\_TMTA\_DEFAULT  (UEM\_TMTA\_ENABLED)

*По умолчанию: Передатчик шины А включен.*

Значения параметра UEM\_TMTB\_DIS.

#define UEM\_TMTB\_DISABLED  1

*Передатчик шины Б отключен.*

#define UEM\_TMTB\_ENABLED  0

*Передатчик шины Б включен.*

#define UEM\_TMTB\_DEFAULT  (UEM\_TMTB\_ENABLED)

*По умолчанию: Передатчик шины Б включен.*

Значения параметра UEM\_RCVA\_DIS.

#define UEM\_RCVA\_DISABLED  1

*Приемник шины А отключен.*

#define UEM\_RCVA\_ENABLED  0

*Приемник шины А включен.*

#define UEM\_RCVA\_DEFAULT  (UEM\_RCVA\_ENABLED)

*По умолчанию: Приемник шины А включен.*

Значения параметра UEM\_RCVB\_DIS.

#define UEM\_RCVB\_DISABLED  1

*Приемник шины Б отключен.*

#define UEM\_RCVB\_ENABLED  0

*Приемник шины Б включен.*

#define UEM\_RCVB\_DEFAULT  (UEM\_RCVB\_ENABLED)

*По умолчанию: Приемник шины Б включен.*

Значения параметра UEM\_SYNC\_IN\_1\_ENA.

#define UEM\_SYNC\_IN\_1\_DISABLED  0

*Обработка входного сигнала синхронизации 1 запрещена.*

#define UEM\_SYNC\_IN\_1\_ENABLED  1

*Обработка входного сигнала синхронизации 1 разрешена.*

#define UEM\_SYNC\_IN\_1\_DEFAULT  (UEM\_SYNC\_IN\_1\_DISABLED)

*По умолчанию: Обработка входного сигнала синхронизации 1 запрещена.*

Значения параметра UEM\_SYNC\_IN\_2\_ENA.

#define UEM\_SYNC\_IN\_2\_DISABLED  0

*Обработка входного сигнала синхронизации 2 запрещена.*

#define UEM\_SYNC\_IN\_2\_ENABLED  1

*Обработка входного сигнала синхронизации 2 разрешена.*

#define UEM\_SYNC\_IN\_2\_DEFAULT  (UEM\_SYNC\_IN\_2\_DISABLED)

*По умолчанию: Обработка входного сигнала синхронизации 2 запрещена.*

Значения параметра UEM\_SYNC\_OUT\_1\_ENA.

#define UEM\_SYNC\_OUT\_1\_DISABLED  0

*Формирование выходного сигнала синхронизации 1 запрещено.*

#define UEM\_SYNC\_OUT\_1\_ENABLED  1

*Формирование выходного сигнала синхронизации 1 разрешено.*

#define UEM\_SYNC\_OUT\_1\_DEFAULT  (UEM\_SYNC\_OUT\_1\_DISABLED)

*По умолчанию: Формирование выходного сигнала синхронизации 1 запрещено.*

Значения параметра UEM\_SYNC\_OUT\_2\_ENA.

#define UEM\_SYNC\_OUT\_2\_DISABLED  0

*Формирование выходного сигнала синхронизации 2 запрещено.*

#define UEM\_SYNC\_OUT\_2\_ENABLED  1

*Формирование выходного сигнала синхронизации 2 разрешено.*

#define UEM\_SYNC\_OUT\_2\_DEFAULT  (UEM\_SYNC\_OUT\_2\_DISABLED)

*По умолчанию: Формирование выходного сигнала синхронизации 2 запрещено.*

Значения параметра UEM\_BRCST\_DIS.

#define UEM\_BRCST\_DISABLED  1

*Групповые сообщения запрещены.*

#define UEM\_BRCST\_ENABLED  0

*Групповые сообщения разрешены.*

#define UEM\_BRCST\_DEFAULT  (UEM\_BRCST\_ENABLED)

*По умолчанию: Групповые сообщения разрешены.*

Значения параметра UEM\_SYNC\_IN\_1\_INTGEN.

#define UEM\_SYNC\_IN\_1\_INTGEN\_DISABLED  0

*Внутренняя генерация запрещена.*

#define UEM\_SYNC\_IN\_1\_INTGEN\_ENABLED  1

*Внутренняя генерация разрешена.*

#define UEM\_SYNC\_IN\_1\_INTGEN\_DEFAULT  (UEM\_SYNC\_IN\_1\_INTGEN\_DISABLED)

*По умолчанию: запрещена.*

Значения параметра UEM\_SYNC\_IN\_2\_INTGEN.

#define UEM\_SYNC\_IN\_2\_INTGEN\_DISABLED  0

*Внутренняя генерация запрещена.*

#define UEM\_SYNC\_IN\_2\_INTGEN\_ENABLED  1

*Внутренняя генерация разрешена.*

#define UEM\_SYNC\_IN\_2\_INTGEN\_DEFAULT  (UEM\_SYNC\_IN\_2\_INTGEN\_DISABLED)

*По умолчанию: запрещена.*

Значения параметра UEM\_ERR\_INJ\_DIS.

#define UEM\_ERR\_INJ\_DISABLED  1

*Внесение ошибок кодирования запрещено.*

#define UEM\_ERR\_INJ\_ENABLED  0

*Внесение ошибок кодирования разрешено.*

#define UEM\_ERR\_INJ\_DEFAULT  (UEM\_ERR\_INJ\_ENABLED)

*По умолчанию: Внесение ошибок кодирования разрешено.*

Значение UEM\_SET.

*Значение* ***UEM\_SET****, тождественно равное 1, используется с параметрами, фактически являющимися командами, вызывающими определенные действия аппаратуры УЭМ. Такие параметры доступны только по записи, причем действенной является именно запись значения 1. Такие параметры не имеют и не требуют установки значений по умолчанию.*

#define **UEM\_SET**  1

*Установка параметра в 1.*

Значения параметра UEM\_DB\_ACT.

*Данный параметр доступен только по чтению, он не имеет и не требует установки значения по умолчанию.*

#define UEM\_DB\_INACTIVE  0

*Нет активности на шине данных.*

#define UEM\_DB\_ACTIVE  1

*Обнаружена активность на шине данных.*

Значения параметров UEM\_TXA\_RFT, UEM\_TXB\_RFT.

#define **UEM\_RFT\_MIN**  0

*Минимальное значение.*

#define **UEM\_RFT\_MAX**  31

*Максимальное значение.*

#define **UEM\_RFT\_SINE**  32

*Специальное значение. Включает формирование синусоидального сигнала.*

#define UEM\_RFT\_DEFAULT  18

*Значение по умолчанию (18).*

Значения параметров UEM\_TXA\_VPP, UEM\_TXB\_VPP.

#define **UEM\_VPP\_MIN**  0x00E

*Минимальное значение.*

#define **UEM\_VPP\_MAX**  0x01FF

*Максимальное значение.*

#define **UEM\_VPP\_DEFAULT**  0x0136

*Значение по умолчанию (0x0136).*

Значения параметра UEM\_MC\_DIS.

#define UEM\_MC\_DISABLED  1

*Команды управления запрещены.*

#define UEM\_MC\_ENABLED  0

*Команды управления разрешены.*

#define UEM\_MC\_DEFAULT  (UEM\_MC\_ENABLED)

*По умолчанию: Команды управления разрешены.*

Значения параметра UEM\_BTMT\_DIS.

#define UEM\_BTMT\_DISABLED  1

*Блокирование и разблокирование передатчиков ОУ запрещено.*

#define UEM\_BTMT\_ENABLED  0

*Блокирование и разблокирование передатчиков ОУ разрешено.*

#define UEM\_BTMT\_DEFAULT  (UEM\_BTMT\_ENABLED)

*По умолчанию: Блокирование и разблокирование передатчиков ОУ разрешено.*

Значения параметра UEM\_BRTF\_DIS.

#define UEM\_BRTF\_DISABLED  1

*Блокирование и разблокирование признака неисправности ОУ запрещено.*

#define UEM\_BRTF\_ENABLED  0

*Блокирование и разблокирование признака неисправности ОУ разрешено.*

#define UEM\_BRTF\_DEFAULT  (UEM\_BRTF\_ENABLED)

*По умолчанию: Блокирование и разблокирование признака неисправности ОУ разрешено.*

Значения параметра UEM\_SYNC2\_VRTA.

#define UEM\_SYNC2\_VRTA\_MIN  0

*Минимальное значение.*

#define UEM\_SYNC2\_VRTA\_MAX  31

*Максимальное значение.*

#define UEM\_SYNC2\_VRTA\_DEFAULT  0

*Значение по умолчанию.*

Значения параметра UEM\_SYNC1\_C\_D\_.

#define UEM\_SYNC1\_ON\_COMMAND  1

*Синхроимпульс командного/ответного слова.*

#define UEM\_SYNC1\_ON\_DATA  0

*Синхроимпульс слова данных.*

#define UEM\_SYNC1\_C\_D\_DEFAULT  (UEM\_SYNC1\_ON\_DATA)

*По умолчанию: Синхроимпульс слова данных.*

Значения параметра UEM\_SYNC1\_ERR.

#define UEM\_SYNC1\_ON\_ERROR  1

*При наличии ошибки в слове.*

#define UEM\_SYNC1\_ON\_NO\_ERROR  0

*При отсутствии ошибки в слове.*

#define UEM\_SYNC1\_ERR\_DEFAULT  (UEM\_SYNC1\_ON\_NO\_ERROR)

*По умолчанию: При отсутствии ошибки в слове.*

Значения параметра UEM\_SYNC1\_GAPB.

#define UEM\_SYNC1\_ON\_GAPB  1

*При наличии паузы перед словом.*

#define UEM\_SYNC1\_ON\_NO\_GAPB  0

*При отсутствии паузы перед словом.*

#define UEM\_SYNC1\_GAPB\_DEFAULT  (UEM\_SYNC1\_ON\_NO\_GAPB)

*По умолчанию: При отсутствии паузы перед словом.*

Значения параметра UEM\_SYNC1\_CH.

#define UEM\_SYNC1\_CH\_A  0

*Слово передается по шине А.*

#define UEM\_SYNC1\_CH\_B  2

*Слово передается по шине Б.*

#define UEM\_SYNC1\_ACH  1

*Слово передается по любой шине.*

#define UEM\_SYNC1\_CH\_DEFAULT  (UEM\_SYNC1\_CH\_A)

*Значение* ***UEM\_SYNC1\_CH*** *по умолчанию: по шине А.*

Значения параметров UEM\_IST1, UEM\_IST2.

#define **UEM\_IST\_MIN**  1

*Минимальное значение.*

#define **UEM\_IST\_MAX**  65536

*Максимальное значение.*

#define UEM\_IST\_DEFAULT  (UEM\_IST\_MAX)

*Значение по умолчанию.*

Значения параметров UEM\_BM\_WORD\_PATTERN, UEM\_BM\_WORD\_MASK.

#define **UEM\_BM\_WORD\_MIN**  0x0000

*Минимальное значение.*

#define **UEM\_BM\_WORD\_MAX**  0xFFFF

*Максимальное значение.*

#define UEM\_BM\_WORD\_DEFAULT  (UEM\_BM\_WORD\_MIN)

*Значение по умолчанию.*

Более мнемоничные обозначения форматов сообщений

#define UEM\_BCRT  UEM\_F1

*КШОУ.*

#define UEM\_RTBC  UEM\_F2

*ОУКШ.*

#define UEM\_RTRT  UEM\_F3

*ОУОУ.*

#define UEM\_MC  UEM\_F4

*Команда управления.*

#define UEM\_MCRTBC  UEM\_F5

*Команда управления со словом данных, передаваемым от ОУ к КШ.*

#define UEM\_MCBCRT  UEM\_F6

*Команда управления со словом данных, передаваемым от КШ к ОУ.*

#define UEM\_BCRTb  UEM\_F7

*КШОУ ГРУППОВОЕ.*

#define UEM\_RTRTb  UEM\_F8

*ОУОУ ГРУППОВОЕ.*

#define UEM\_MCb  UEM\_F9

*Команда управления групповая.*

#define UEM\_MCBCRTb  UEM\_F10

*Команда управления групповая со словом данных, передаваемым от КШ к ОУ.*

Число слов данных

#define UEM\_NDATA\_MIN  0

*Число слов данных - минимальное значение.*

#define UEM\_NDATA\_MAX  62

*Число слова данных - максимальное значение.*

#define UEM\_NDATA\_BY\_CW  63

*Число слов данных определяется командным словом.*

Признаки ошибок распознавания сообщения.

*Комбинации перечисленных констант (по |) определяют содержание элементов типа* ***UEM\_ERROR\_FLAGS****.*

#define UEM\_ERRF\_ERROR  1

*Наличие любой ошибки (суммарный флаг).*

#define UEM\_ERRF\_ENCODING  (1<<1)

*Наличие ошибки кодирования слов (суммарный флаг).*

#define **UEM\_ERRF\_FORMAT**  (1<<2)

*Наличие нарушения формата, ошибки состава сообщения (суммарный флаг).*

#define **UEM\_ERRF\_MINGAP**  (1<<3)

*Временной интервал меньше допустимого.*

#define UEM\_ERRF\_NO\_RESPONSE  (1<<4)

*Отсутствие ответа.*

#define UEM\_ERRF\_SYNC\_TYPE  (1<<5)

*Неверный тип синхроимпульса.*

#define UEM\_ERRF\_MISSING\_CWSW  (1<<6)

*Отсутствует командное или ответное слово.*

#define UEM\_ERRF\_EXTRA\_CWSW  (1<<7)

*Лишнее командное или ответное слово.*

#define UEM\_ERRF\_MISSING\_DW  (1<<8)

*Недостаточно слов данных.*

#define UEM\_ERRF\_EXTRA\_DW  (1<<9)

*Лишние слова данных.*

#define UEM\_ERRF\_INCORRECT\_RTN  (1<<10)

*Некорректный адрес ОУ.*

#define UEM\_ERRF\_RTRT\_FORMAT  (1<<11)

*Ошибка формата ОУОУ (одинаковые адреса ОУ, несовпадение числа СД).*

#define UEM\_ERRF\_INC\_MODE\_CODE  (1<<12)

*Некорректная команда управления.*

#define UEM\_ERRF\_FORMAT\_MC  (1<<13)

*Ошибка формата команды управления.*

#define **UEM\_ERRF\_GAPN**  (1<<20)

*Недостоверная информация (сигнал) во время паузы перед словом.*

#define **UEM\_ERRF\_PARITY**  (1<<21)

*Ошибка четности.*

#define UEM\_ERRF\_LESS\_BITS  (1<<22)

*Укороченное слово.*

#define UEM\_ERRF\_MORE\_BITS  (1<<23)

*Удлиненное слово.*

#define **UEM\_ERRF\_ENC**  (1<<24)

*Ошибка бифазного кодирования.*

#define **UEM\_ERRF\_DT**  (1<<30)

*Несоблюдение минимальной паузы перед словом, по данным аппаратного декодера.*

#define UEM\_ERRF\_ENCODING2  (1<<31)

*Наличие любой ошибки кодирования слов (суммарный флаг, по данным аппаратного декодера).*

Флаги отсчета паузы.

*Пауза отсчитывается от одного из определенных событий.*

*Флаги отсчета паузы определяют выбор события.*

*Пауза может отсчитываться от начала либо конца предшествующего сегмента, переданного КШ либо другим абонентом (ОУ), по той же либо по альтернативной шине. Выбор каждой из альтернатив управляется отдельным флагом. Для начала отсчета паузы также может использоваться сигнал внешней синхронизации* ***sync\_in\_1****. Поступление этого сигнала используется вместо события начала или конца сегмента, переданного КШ. Если же задан отсчет паузы от начала (или конца) сегмента, переданного другим абонентом, то учитываются оба события - начало (или конец) сегмента и поступление сигнала. При задании отсчета паузы от начала (или конца) сегмента другого абонента, или от поступления сигнала, или обоих, то есть - от внешних событий, которые теоретически могут и не наступить, используется таймаут, который задается в функции* ***uem\_cseg\_gap\_set()****, и по истечении которого отсчет паузы начинается без дальнейшего ожидания внешних событий.*

#define UEM\_CSEG\_GAP\_FROM\_START  0x20

*Отсчет паузы от начала предыдущего сегмента.*

#define UEM\_CSEG\_GAP\_FROM\_END  0

*Отсчет паузы от конца предыдущего сегмента.*

#define UEM\_CSEG\_GAP\_ESYNC  0x10

*Отсчет паузы после сигнала внешней синхронизации (только для КШ).*

#define UEM\_CSEG\_GAP\_ALT\_BUS  0x08

*Отсчет паузы от сегмента по альтернативной шине (только для КШ).*

#define UEM\_CSEG\_GAP\_THIS\_BUS  0

*Отсчет паузы от сегмента по этой же шине (только для КШ).*

#define UEM\_CSEG\_GAP\_ALT\_AB  0x04

*Отсчет паузы от сегмента другого абонента (ОУ) (только для КШ).*

#define UEM\_CSEG\_GAP\_THIS\_AB  0

*Отсчет паузы от собственного сегмента (только для КШ).*

#define UEM\_CSEG\_GAP\_DEFAULT\_FLAGS  (UEM\_CSEG\_GAP\_FROM\_START)

*Стандартный набор флагов отсчета паузы.*

Диапазоны значений параметров отсчета паузы.

#define UEM\_CSEG\_GAP\_MIN  0

*Минимальное значение паузы.*

#define UEM\_CSEG\_GAP\_MAX  65535

*Максимальное значение паузы.*

#define UEM\_CSEG\_GAP\_DEFAULT\_VALUE  0

*Значение паузы по умолчанию.*

#define UEM\_CSEG\_GAP\_TIMEOUT\_MIN  0

*Минимальное значение таймаута отсчета паузы.*

#define UEM\_CSEG\_GAP\_TIMEOUT\_MAX  1023

*Максимальное значение таймаута отсчета паузы.*

#define UEM\_CSEG\_GAP\_DEFAULT\_TIMEOUT  0

*Таймаут отсчета паузы по умолчанию.*

Константы для числа повторов кадра.

*Данные константы могут использоваться в аргументе* ***repeat\_count*** *функции* ***uem\_bcp\_append\_frame()****.*

#define UEM\_FRAME\_REPEAT\_UNLIM  0

*Неограниченное число повторов кадра.*

#define UEM\_FRAME\_REPEAT\_MIN  1

*Минимальное число повторов кадра.*

#define UEM\_FRAME\_REPEAT\_MAX  1023

*Максимальное число повторов кадра.*

#define UEM\_FRAME\_REPEAT\_DEFAULT  (UEM\_FRAME\_REPEAT\_MIN)

*Число повторов кадра по умолчанию (1).*

Флаги кадра.

*Данные константы могут использоваться в аргументе* ***frame\_flags*** *функции* ***uem\_bcp\_append\_frame()****.*

#define **UEM\_FRAME\_STOP**  0x0001

*Остановка КШ.*

#define **UEM\_FRAME\_ALLRPT**  0x0002

*Зацикливание программы КШ.*

#define UEM\_FRAME\_NONE  0

*Нет указаний.*

#define UEM\_FRAME\_CONT  (UEM\_FRAME\_NONE)

*Продолжение программы КШ.*

#define UEM\_FRAME\_DEFAULT  (UEM\_FRAME\_STOP)

*Флаги кадра по умолчанию: остановка КШ.*

Виды размерностей программы КШ.

#define UEM\_BCP\_NFRAMES  (-1)

*Число кадров.*

#define UEM\_BCP\_CUR\_SIZE  (-2)

*Текущий размер.*

#define UEM\_BCP\_MAX\_SIZE  (-3)

*Максимальный размер.*

Признаки обработки командного слова в ОУ.

*Из объединения (по |) этих констант составляется аргумент* ***rtdes*** *в функциях* ***uem\_response\_create()****,* ***uem\_response\_read()****.*

#define **UEM\_RTDES\_SW\_DIS**  (1<<15)

*Не отвечать.*

#define **UEM\_RTDES\_DBCA**  (1<<6)

*Принять управление интерфейсом.*

#define UEM\_RTDES\_DBCA\_BCSTART  (1<<7)

*Запустить КШ.*

#define UEM\_RTDES\_COM\_ILLEGAL  (1<<17)

*Недопустимая команда.*

#define **UEM\_RTDES\_LCMD\_DW**  (1<<13)

*Передать последнюю команду.*

#define **UEM\_RTDES\_SWB\_SAV**  (1<<12)

*Автоматическое формирование флагов ОС.*

#define UEM\_RTDES\_WRONG\_CH  (1<<14)

*Отвечать по другой шине.*

#define **UEM\_RTDES\_WA**  (1<<8)

*Циркулярный возврат данных.*

#define UEM\_RTDES\_WA\_BRCST  (1<<31)

*Циркулярный возврат в групповых командах.*

#define UEM\_RTDES\_ILLEG\_MASK  (1<<16)

*Задать маску допустимых команд в зависимости от количества слов данных.*

#define UEM\_RTDES\_DEFAULT  (UEM\_RTDES\_SWB\_SAV)

*Значение параметра* ***rtdes*** *по умолчанию.*

Вспомогательные константы для полей командных слов.

*В соответствии с [****1****].*

#define UEM\_RTADDR\_MIN  0

*Минимальное значение адреса ОУ.*

#define UEM\_RTADDR\_MАХ  30

*Максимальное значение адреса ОУ.*

#define UEM\_RTADDR\_BRCST  31

*Адрес для групповых сообщений.*

#define UEM\_RTADDR\_MAX\_EXT  31

*Максимальное значение адреса ОУ в сетях с запретом групповых сообщений.*

#define **UEM\_RT\_RX**  0

*Признак передача/прием: прием.*

#define **UEM\_RT\_TX**  1

*Признак передача/прием: передача.*

#define UEM\_SADDR\_MIN  1

*Минимальное значение подадреса.*

#define UEM\_SADDR\_MAX  30

*Максимальное значение подадреса.*

#define UEM\_SADDR\_CONV\_LOOPBACK  30

*Традиционный подадрес для тестовой петли.*

#define UEM\_SADDR\_MIN\_EXT  0

*Минимальное значение подадреса в сетях с запретом команд управления.*

#define UEM\_SADDR\_MAX\_EXT  31

*Максимальное значение подадреса в сетях с запретом команд управления.*

#define **UEM\_MODE\_0**  0

*Код режима 0.*

#define **UEM\_MODE\_31**  31

*Код режима 31.*

Коды команд управления.

*В соответствии с [****1****].*

#define UEM\_MCODE\_ADBC  0

*Принять управление интерфейсом.*

#define UEM\_MCODE\_SYNCHRO  1

*Синхронизация.*

#define UEM\_MCODE\_TXSTATUS  2

*Передать ОС.*

#define UEM\_MCODE\_BSELFTEST  3

*Начать самоконтроль.*

#define UEM\_MCODE\_BTMT  4

*Блокировать передатчик.*

#define UEM\_MCODE\_UBTMT  5

*Разблокировать передатчик.*

#define UEM\_MCODE\_BRTF  6

*Блокировать признак неисправности ОУ.*

#define UEM\_MCODE\_UBRTF  7

*Разблокировать признак неисправности ОУ.*

#define UEM\_MCODE\_RESETRT  8

*Установить ОУ в исходное состояние.*

#define UEM\_MCODE\_TXVECT  16

*Передать векторное слово.*

#define UEM\_MCODE\_SYNCHRO\_D  17

*Синхронизация (с СД).*

#define UEM\_MCODE\_TXLCMD  18

*Передать последнюю команду.*

#define UEM\_MCODE\_TXBIT  19

*Передать слово ВСК ОУ.*

#define UEM\_MCODE\_BTMT\_I  20

*Блокировать i-й передатчик.*

#define UEM\_MCODE\_UBTMT\_I  21

*Разблокировать i-й передатчик.*

Признаки ответного слова.

*В соответствии с [****1****].*

#define **UEM\_RTFAIL**  0x0001

*НОУ - Неисправность оконечного устройства.*

#define **UEM\_DBCA**  0x0002

*ПУИ - Принято управление интерфейсом.*

#define **UEM\_ABFAIL**  0x0004

*НА - Неисправность абонента.*

#define **UEM\_ABBUSY**  0x0008

*АЗ - Абонент занят.*

#define **UEM\_BCCA**  0x0010

*ПГК - Принята групповая команда.*

#define **UEM\_RSV14**  0x0020

*Резервный бит 14.*

#define **UEM\_RSV13**  0x0040

*Резервный бит 13.*

#define **UEM\_RSV12**  0x0080

*Резервный бит 12.*

#define **UEM\_SERVRQ**  0x0100

*ЗО - Запрос обслуживания.*

#define **UEM\_ZERO**  0x0200

*Передача ОС.*

#define **UEM\_MSGERR**  0x0400

*ОШС - Ошибка в сообщении.*

### Определения типов

* typedef ViSession **UEM\_DEVHANDLE**

*Дескриптор устройства УЭМ или виртуального устройства в составе УЭМ.*

* typedef ViSession **UEM\_OBJHANDLE**

*Дескриптор объекта в ОЗУ УЭМ.*

* typedef ViBoolean **UEM\_BOOL**

*Логическое значение.*

* typedef ViUInt16 **UEM\_PARAMID**

*Идентификатор параметра.*

* typedef ViUInt32 **UEM\_DWORD**

*32-битное целое без знака.*

* typedef ViUInt16 **UEM\_WORD**

*16-битное целое без знака.*

* typedef ViUInt64 **UEM\_TIME\_TAG\_LIN**

*Метка времени в линейном формате*

* typedef ViUInt32 **UEM\_ERROR\_FLAGS**

***Признаки ошибок распознавания сообщения*** *в мониторе шины.*

* typedef void(\* **uem\_bm\_handler**) (**UEM\_DEVHANDLE** bm, void \*userdata)

*Обработчик события МШ.*

### Перечисления

* enum **UEM\_ERROR\_TYPE** { **UEM\_ERRT\_NONE** = 0, **UEM\_ERRT\_INV\_PARITY** = 1, **UEM\_ERRT\_WRONG\_BITCOUNT** = 2, **UEM\_ERRT\_BAD\_SYNCHRO** = 3, **UEM\_ERRT\_BAD\_BIPHASE\_ZERO** = 4, **UEM\_ERRT\_BAD\_BIPHASE\_POS** = 5, **UEM\_ERRT\_BAD\_BIPHASE\_NEG** = 6, **UEM\_ERRT\_SHIFT\_EDGE** = 7 }

*Тип вносимой ошибки кодирования.*

* enum **UEM\_SYNCHRO\_ERROR\_POS** { **UEM\_BAD\_SYNCHRO\_NONE** = 0, **UEM\_BAD\_SYNCHRO\_IEEEEE** = 1, **UEM\_BAD\_SYNCHRO\_EIEEEE** = 2, **UEM\_BAD\_SYNCHRO\_EEIEEE** = 3, **UEM\_BAD\_SYNCHRO\_NONE2** = 4, **UEM\_BAD\_SYNCHRO\_EEEIEE** = 5, **UEM\_BAD\_SYNCHRO\_EEEEIE** = 6, **UEM\_BAD\_SYNCHRO\_EEEEEI** = 7 }

*Позиция ошибки кодирования синхроимпульса.*

* enum **UEM\_HANDLE\_TYPE** { **UEM\_INVH**, **UEM\_UEM**, **UEM\_BC**, **UEM\_RT**, **UEM\_BM**, **UEM\_CSEG**, **UEM\_BCP**, **UEM\_RESP** }

*Тип дескриптора объекта библиотеки УЭМ.*

* enum **UEM\_TIME\_PARAM** { **UEM\_MIN\_T1**, **UEM\_MIN\_T2**, **UEM\_RTMO** }

*Идентификатор (селектор) параметра интервала времени*

* enum **UEM\_FORMAT** { **UEM\_UNF**, **UEM\_F1**, **UEM\_F2**, **UEM\_F3**, **UEM\_F4**, **UEM\_F5**, **UEM\_F6**, **UEM\_F7**, **UEM\_F8**, **UEM\_F9**, **UEM\_F10** }

*Форматы сообщений (номера по ГОСТ [****1****]).*

* enum **UEM\_CHANNEL** { **UEM\_CH\_A**, **UEM\_CH\_B** }

*Селектор шины (А/Б).*

* enum **UEM\_SYNC** { **UEM\_SYNC\_D**, **UEM\_SYNC\_C** }

*Селектор синхроимпульса.*

* enum **UEM\_CSEG\_TYPE** { **UEM\_CSEG\_NORMAL**, **UEM\_CSEG\_OVERLAY**, **UEM\_CSEG\_GAP** }

*Тип командного сегмента.*

### Функции

* ViStatus **uem\_init** (ViRsrc idstr, ViBoolean idn, ViBoolean reset, ViSession \*uem)

*Инициализация объекта УЭМ.*

* ViStatus **uem\_connect** (ViSession uem, ViSession vi, ViUInt16 meznum, ViBoolean idn, ViBoolean reset)

*Привязка объекта УЭМ к сеансу носителя мезонина.*

* ViStatus **uem\_bc\_init** (**UEM\_DEVHANDLE** \*bc, **UEM\_DEVHANDLE** uem)

*Открытие виртуального КШ в составе УЭМ.*

* ViStatus **uem\_rt\_init** (**UEM\_DEVHANDLE** \*rt, **UEM\_DEVHANDLE** uem, **UEM\_WORD** rtaddr)

*Открытие виртуального ОУ в составе УЭМ.*

* ViStatus **uem\_bm\_init** (**UEM\_DEVHANDLE** \*bm, **UEM\_DEVHANDLE** uem)

*Открытие виртуального МШ в составе УЭМ.*

* ViStatus **uem\_close** (**UEM\_DEVHANDLE** anydev)

*Закрытие УЭМ или любого виртуального устройства в составе УЭМ.*

* **UEM\_HANDLE\_TYPE** **uem\_handle\_type** (**UEM\_DEVHANDLE** anyobject)

*Тип дескриптора объекта библиотеки УЭМ.*

* ViStatus **uem\_parent\_dev** (**UEM\_DEVHANDLE** anyobject, **UEM\_DEVHANDLE** \*parentdev)

*Родительское устройство.*

* ViStatus **uem\_root\_dev** (**UEM\_DEVHANDLE** anyobject, **UEM\_DEVHANDLE** \*uem)

*Физическое устройство УЭМ.*

* ViStatus **uem\_layer\_handle** (**UEM\_DEVHANDLE** uem, ViUInt32 sel, ViSession \*handle)

*Связь с ПО нижележащих слоев.*

* ViStatus **uem\_param\_get** (**UEM\_DEVHANDLE** uem, **UEM\_PARAMID** paramid, **UEM\_DWORD** \*value)

*Считывание значения конфигурационного параметра.*

* ViStatus **uem\_param\_set** (**UEM\_DEVHANDLE** uem, **UEM\_PARAMID** paramid, **UEM\_DWORD** value)

*Запись значения конфигурационного параметра.*

* ViStatus **uem\_timing\_set** (**UEM\_DEVHANDLE** anydev, **UEM\_TIME\_PARAM** param, **UEM\_DWORD** value)

*Установка параметра интервала времени.*

* ViStatus **uem\_timing\_get** (**UEM\_DEVHANDLE** anydev, **UEM\_TIME\_PARAM** param, **UEM\_DWORD** \*value)

*Считывание параметра интервала времени.*

* ViStatus **uem\_time\_tag\_get** (**UEM\_DEVHANDLE** uem, **UEM\_TIME\_TAG** \*time\_tag)

*Считывание встроенного счетчика времени.*

* ViStatus **uem\_time\_tag\_set** (**UEM\_DEVHANDLE** uem, **UEM\_TIME\_TAG** \*time\_tag)

*Установка значения встроенного счетчика времени.*

* ViStatus **uem\_time\_tag\_reset** (**UEM\_DEVHANDLE** uem)

*Сброс встроенного счетчика времени.*

* **UEM\_TIME\_TAG\_LIN** **uem\_time\_tag\_to\_linear** (**UEM\_TIME\_TAG** \*time\_tag)

*Перевод метки времени в линейный формат.*

* void **uem\_time\_tag\_to\_struct** (**UEM\_TIME\_TAG** \*time\_tag, **UEM\_TIME\_TAG\_LIN** linear)

*Перевод метки времени из линейного в структурированный формат.*

* ViStatus **uem\_bc\_cseg\_format** (**UEM\_DEVHANDLE** bc, **UEM\_CMD\_SEG** \*cseg\_data, **UEM\_CHANNEL** ch, **UEM\_FORMAT** format, **UEM\_WORD** rt, **UEM\_WORD** sa, **UEM\_WORD** ndatawords, **UEM\_WORD** \*datawords)

*Формирование образов командных сегментов для сообщений форматов 1,2,7 и неформатных сообщений.*

* ViStatus **uem\_bc\_cseg\_format\_RTRT** (**UEM\_DEVHANDLE** bc, **UEM\_CMD\_SEG** \*cseg\_data, **UEM\_CHANNEL** ch, **UEM\_FORMAT** format, **UEM\_WORD** rtrx, **UEM\_WORD** sarx, **UEM\_WORD** rttx, **UEM\_WORD** satx, **UEM\_WORD** ndatawords)

*Формирование образов командных сегментов для сообщений форматов 3,8.*

* ViStatus **uem\_bc\_cseg\_format\_MODE** (**UEM\_DEVHANDLE** bc, **UEM\_CMD\_SEG** \*cseg\_data, **UEM\_CHANNEL** ch, **UEM\_FORMAT** format, **UEM\_WORD** rt, **UEM\_WORD** mode, **UEM\_WORD** modecode, **UEM\_WORD** dataword)

*Формирование образов командных сегментов для сообщений форматов 4,5,6,9,10.*

* ViStatus **uem\_bc\_cseg\_create** (**UEM\_DEVHANDLE** bc, **UEM\_OBJHANDLE** \*cseg, **UEM\_CMD\_SEG** \*cseg\_data)

*Создание командного сегмента.*

* ViStatus **uem\_cseg\_read** (**UEM\_OBJHANDLE** cseg, **UEM\_CMD\_SEG** \*cseg\_data)

*Чтение командного сегмента.*

* ViStatus **uem\_cseg\_gap\_set** (**UEM\_OBJHANDLE** cseg, **UEM\_WORD** gap, **UEM\_WORD** gap\_flags, **UEM\_WORD** gap\_timeout)

*Программирование паузы перед сообщением.*

* ViStatus **uem\_cseg\_gap\_get** (**UEM\_OBJHANDLE** cseg, **UEM\_WORD** \*gap, **UEM\_WORD** \*gap\_flags, **UEM\_WORD** \*gap\_timeout)

*Считывание паузы перед сообщением.*

* ViStatus **uem\_cseg\_gap\_reset** (**UEM\_OBJHANDLE** cseg)

*Сброс паузы перед сообщением.*

* ViStatus **uem\_cseg\_word\_gap\_set** (**UEM\_OBJHANDLE** cseg, **UEM\_WORD** wordnumber, **UEM\_WORD** gap, **UEM\_WORD** gap\_flags, **UEM\_WORD** gap\_timeout)

*Программирование паузы между словами.*

* ViStatus **uem\_cseg\_word\_gap\_get** (**UEM\_OBJHANDLE** cseg, **UEM\_WORD** wordnumber, **UEM\_WORD** \*gap, **UEM\_WORD** \*gap\_flags, **UEM\_WORD** \*gap\_timeout)

*Считывание паузы перед словом.*

* ViStatus **uem\_cseg\_error\_set** (**UEM\_OBJHANDLE** cseg, **UEM\_WORD** wordnumber, **UEM\_ERROR\_TYPE** error\_type, ViInt32 error\_pos, ViInt32 error\_param)

*Внесение ошибок кодирования.*

* ViStatus **uem\_cseg\_error\_get** (**UEM\_OBJHANDLE** cseg, **UEM\_WORD** wordnumber, **UEM\_ERROR\_TYPE** \*error\_type, ViInt32 \*error\_pos, ViInt32 \*error\_param)

*Считывание внесенных ошибок кодирования.*

* ViStatus **uem\_cseg\_sync\_set** (**UEM\_OBJHANDLE** cseg, **UEM\_WORD** wordnumber, **UEM\_SYNC** sync)

*Установка типа синхроимпульса.*

* ViStatus **uem\_cseg\_sync\_get** (**UEM\_OBJHANDLE** cseg, **UEM\_WORD** wordnumber, **UEM\_SYNC** \*sync)

*Считывание типа синхроимпульса.*

* ViStatus **uem\_bc\_gap\_create** (**UEM\_DEVHANDLE** bc, **UEM\_OBJHANDLE** \*cseg, **UEM\_WORD** gap, **UEM\_WORD** gap\_flags, **UEM\_WORD** gap\_timeout, **UEM\_CHANNEL** ch)

*Создание паузы.*

* ViStatus **uem\_bc\_cseg\_overlay** (**UEM\_DEVHANDLE** bc, **UEM\_OBJHANDLE** \*cseg\_o, **UEM\_OBJHANDLE** cseg\_1, **UEM\_WORD** gap, **UEM\_OBJHANDLE** cseg\_2)

*Создание сообщения с наложением.*

* ViStatus **uem\_cseg\_type** (**UEM\_OBJHANDLE** cseg, **UEM\_CSEG\_TYPE** \*type)

*Запрос типа командного сегмента.*

* ViStatus **uem\_cseg\_desrtoy** (**UEM\_OBJHANDLE** cseg)

*Уничтожение командного сегмента.*

* ViStatus **uem\_bcp\_create** (**UEM\_OBJHANDLE** \*bcprog, **UEM\_DWORD** max\_size, **UEM\_DEVHANDLE** bc)

*Создание программы КШ.*

* ViStatus **uem\_bcp\_append\_frame** (**UEM\_OBJHANDLE** bcprog, **UEM\_WORD** repeat\_count, **UEM\_WORD** frame\_flags, int \*frameindex)

*Добавление кадра в конец программы КШ.*

* ViStatus **uem\_bcp\_append\_cseg** (**UEM\_OBJHANDLE** bcprog, **UEM\_OBJHANDLE** cseg, int \*csegindex)

*Добавление командного сегмента в конец кадра.*

* ViStatus **uem\_bcp\_discover\_cseg** (**UEM\_OBJHANDLE** bcprog, int frameindex, int csegindex, **UEM\_OBJHANDLE** \*cseg)

*Выяснение командного сегмента.*

* ViStatus **uem\_bcp\_replace\_cseg** (**UEM\_OBJHANDLE** bcprog, int frameindex, int csegindex, **UEM\_OBJHANDLE** cseg)

*Замена командного сегмента в кадре.*

* ViStatus **uem\_bcp\_dimension** (**UEM\_OBJHANDLE** bcprog, int frameindex, int \*dim)

*Запрос размерностей программы КШ.*

* ViStatus **uem\_bcp\_inspect\_frame** (**UEM\_OBJHANDLE** bcprog, int frameindex, **UEM\_WORD** \*repeat\_count, **UEM\_WORD** \*frame\_flags)

*Запрос характеристик кадра.*

* ViStatus **uem\_bcp\_install** (**UEM\_OBJHANDLE** bcprog)

*Установка программы КШ в качестве исполняемой.*

* ViStatus **uem\_bcp\_desrtoy** (**UEM\_OBJHANDLE** bcprog)

*Уничтожение объекта "программа КШ" в ОЗУ КШ.*

* ViStatus **uem\_bcp\_set\_standard\_gaps** (**UEM\_OBJHANDLE** bcprog)

*Расчет и установка стандартных пауз между сообщениями (необязательно).*

* ViStatus **uem\_bc\_send\_receive** (**UEM\_DEVHANDLE** bc, **UEM\_CMD\_SEG** \*cseg\_data, **UEM\_BM\_MESSAGE** \*\*msg\_and\_resp)

*Передача отдельного сообщения и получение ответа на него.*

* ViStatus **uem\_response\_create** (**UEM\_OBJHANDLE** \*resp, **UEM\_DEVHANDLE** rt, **UEM\_DWORD** rtdes, **UEM\_DWORD** illeg\_mask, **UEM\_WORD** status, **UEM\_WORD** ndatawords, **UEM\_WORD** \*data)

*Создание ответного сегмента.*

* ViStatus **uem\_response\_read** (**UEM\_OBJHANDLE** resp, **UEM\_DWORD** \*rtdes, **UEM\_DWORD** \*illeg\_mask, **UEM\_WORD** \*status, **UEM\_WORD** \*ndatawords, **UEM\_WORD** \*data)

*Считывание ответного сегмента.*

* ViStatus **uem\_response\_gap\_set** (**UEM\_OBJHANDLE** resp, **UEM\_WORD** gap)

*Установка паузы перед передачей ответного сегмента.*

* ViStatus **uem\_response\_gap\_get** (**UEM\_OBJHANDLE** resp, **UEM\_WORD** \*gap)

*Считывание паузы перед передачей ответного сегмента.*

* ViStatus **uem\_response\_word\_gap\_set** (**UEM\_OBJHANDLE** resp, **UEM\_WORD** wordnumber, **UEM\_WORD** gap)

*Установка паузы перед передачей слова ответного сегмента.*

* ViStatus **uem\_response\_word\_gap\_get** (**UEM\_OBJHANDLE** resp, **UEM\_WORD** wordnumber, **UEM\_WORD** \*gap)

*Считывание паузы перед передачей слова ответного сегмента.*

* ViStatus **uem\_response\_error\_set** (**UEM\_OBJHANDLE** resp, **UEM\_WORD** wordnumber, **UEM\_ERROR\_TYPE** error\_type, ViInt32 error\_pos, ViInt32 error\_param)

*Внесение ошибок кодирования в ответный сегмент.*

* ViStatus **uem\_response\_error\_get** (**UEM\_OBJHANDLE** resp, **UEM\_WORD** wordnumber, **UEM\_ERROR\_TYPE** \*error\_type, ViInt32 \*error\_pos, ViInt32 \*error\_param)

*Считывание ошибок кодирования из ответного сегмента.*

* ViStatus **uem\_response\_sync\_set** (**UEM\_OBJHANDLE** resp, **UEM\_WORD** wordnumber, **UEM\_SYNC** sync)

*Установка типа синхроимпульса.*

* ViStatus **uem\_response\_sync\_get** (**UEM\_OBJHANDLE** resp, **UEM\_WORD** wordnumber, **UEM\_SYNC** \*sync)

*Считывание типа синхроимпульса.*

* ViStatus **uem\_rt\_install\_response** (**UEM\_DEVHANDLE** rt, **UEM\_BOOL** transmit, **UEM\_WORD** sa, **UEM\_OBJHANDLE** resp)

*Установка ответного сегмента как ответа на команду передачи данных.*

* ViStatus **uem\_rt\_install\_response\_MODE** (**UEM\_DEVHANDLE** rt, **UEM\_BOOL** transmit, **UEM\_WORD** mode, **UEM\_WORD** modecode, **UEM\_OBJHANDLE** resp)

*Установка ответного сегмента как ответа на команду управления.*

* ViStatus **uem\_rt\_discover\_response** (**UEM\_DEVHANDLE** rt, **UEM\_BOOL** transmit, **UEM\_WORD** sa, **UEM\_OBJHANDLE** \*resp)

*Выяснение ответа на команду передачи данных.*

* ViStatus **uem\_rt\_discover\_response\_MODE** (**UEM\_DEVHANDLE** rt, **UEM\_BOOL** transmit, **UEM\_WORD** mode, **UEM\_WORD** modecode, **UEM\_OBJHANDLE** \*resp)

*Выяснение ответа на команду управления.*

* ViStatus **uem\_response\_destroy** (**UEM\_OBJHANDLE** resp)

*Уничтожение объекта ответного сегмента в ОЗУ ОУ.*

* ViStatus **uem\_bm\_receive** (**UEM\_DEVHANDLE** bm, **UEM\_BM\_MESSAGE** \*\*message\_data)

*Считывание очередного сообщения, принятого монитором шины.*

* ViStatus **uem\_bm\_queue\_count** (**UEM\_DEVHANDLE** bm, **UEM\_DWORD** \*count)

*Запрос размера очереди сообщений, принятых монитором.*

* ViStatus **uem\_bm\_install\_handler** (**UEM\_DEVHANDLE** bm, **uem\_bm\_handler** handler, void \*userdata)

*Установка обработчика события МШ.*

* ViStatus **uem\_start** (**UEM\_DEVHANDLE** anydev)

*Запуск любого виртуального устройства в составе УЭМ.*

* ViStatus **uem\_stop** (**UEM\_DEVHANDLE** anydev)

*Остановка любого виртуального устройства в составе УЭМ.*

* ViStatus **uem\_is\_running** (**UEM\_DEVHANDLE** anydev, **UEM\_BOOL** \*running)

*Проверка активности виртуального устройства.*

* ViStatus **uem\_error\_message** (**UEM\_DEVHANDLE** uem, ViStatus status, ViChar msg[])

*Запрос сообщения об ошибке.*

* ViStatus **uem\_error\_query** (ViSession uem, ViInt32 \*status, ViChar msg[])

*Запрос последней ошибки.*

* ViStatus **uem\_reset** (**UEM\_DEVHANDLE** anydev)

*Сброс УЭМ или любого виртуального устройства в составе УЭМ.*

* ViStatus **uem\_revision\_query** (**UEM\_DEVHANDLE** uem, ViChar dv[], ViChar iv[])

*Запрос версии.*

* ViStatus **uem\_self\_test** (**UEM\_DEVHANDLE** uem, ViInt16 \*res, ViChar msg[])

*Самоконтроль.*

* **UEM\_WORD** **uem\_command\_word** (**UEM\_WORD** rt, **UEM\_BOOL** tx, **UEM\_WORD** sa, **UEM\_WORD** ndatawords)

*Формирование командного слова.*

* void **uem\_command\_word\_parse** (**UEM\_WORD** cw, **UEM\_WORD** \*rt, **UEM\_BOOL** \*tx, **UEM\_WORD** \*sa, **UEM\_WORD** \*ndatawords)

*Разбор командного слова.*

* **UEM\_WORD** **uem\_status\_word** (**UEM\_WORD** rt, **UEM\_WORD** status\_bits)

*Формирование ответного слова.*

* void **uem\_status\_word\_parse** (**UEM\_WORD** sw, **UEM\_WORD** \*rt, **UEM\_WORD** \*status\_bits)

*Разбор ответного слова.*

### Запуск виртуального КШ с дополнительными параметрами

* #define **UEM\_BC\_START\_NOW**  0

*Нормальный (немедленный) старт.*

* #define **UEM\_BC\_START\_WAITING**  1

*Переход в режим ожидания, старт по внешнему сигналу или команде.*

* #define **UEM\_BC\_START\_DEFAULT**  (**UEM\_BC\_START\_NOW**)

*Стандартный способ старта.*

* enum **UEM\_BC\_STATE\_EX** { **UEM\_BC\_STOPPED**, **UEM\_BC\_WAITING**, **UEM\_BC\_RUNNING** }

*Расширенное состояние КШ.*

* ViStatus **uem\_bc\_start** (**UEM\_DEVHANDLE** bc, **UEM\_DWORD** flags)

*Запуск виртуального КШ в составе УЭМ с дополнительными параметрами.*

* ViStatus **uem\_bc\_state\_ex** (**UEM\_DEVHANDLE** bc, **UEM\_BC\_STATE\_EX** \*state)

*Запрос расширенного состояния КШ.*

### Остановка виртуального КШ с дополнительными параметрами

* #define **UEM\_BC\_STOP\_NOW**  0

*Нормальная (немедленная) остановка.*

* #define **UEM\_BC\_STOP\_ON\_FRAME**  1

*Остановка по завершению текущего кадра.*

* #define **UEM\_BC\_STOP\_DEFAULT**  (**UEM\_BC\_STOP\_NOW**)

*Стандартный способ остановки.*

* ViStatus **uem\_bc\_stop** (**UEM\_DEVHANDLE** bc, **UEM\_DWORD** flags)

*Остановка виртуального КШ в составе УЭМ с дополнительными параметрами.*

### Подробное описание

Универсальные электронные модули УЭМ-МК, МВ98.03. Расширенная библиотека функций. Файл заголовков функций.

# Алфавитный указатель

# 

b

UEM\_TIME\_TAG, 129

ch

UEM\_CMD\_SEG, 121

command1

UEM\_CMD\_SEG, 121

command2

UEM\_CMD\_SEG, 121

cs

UEM\_BM\_MESSAGE, 119

cs\_d

UEM\_BM\_MESSAGE, 119

data

UEM\_CMD\_SEG, 122

UEM\_DATA, 123

UEM\_RAW\_BM\_MESSAGE, 124

UEM\_RESP\_SEG, 125

days

UEM\_TIME\_TAG, 128

dwoffset

UEM\_SEGMENT\_DESCR, 127

end\_time

UEM\_SEGMENT\_DESCR, 126

endoffset

UEM\_SEGMENT\_DESCR, 127

errors

UEM\_BM\_MESSAGE, 120

UEM\_SEGMENT\_DESCR, 126

format

UEM\_CMD\_SEG, 121

gap

UEM\_BM\_MESSAGE, 120

i

UEM\_TIME\_TAG, 129

lostp

UEM\_BM\_MESSAGE, 120

msg\_d

UEM\_BM\_MESSAGE, 119

ndata

UEM\_DATA, 123

offset

UEM\_SEGMENT\_DESCR, 126

overlay

UEM\_BM\_MESSAGE, 120

p\_pr

UEM\_BM\_MESSAGE, 120

quartas

UEM\_TIME\_TAG, 128

raw

UEM\_BM\_MESSAGE, 120

reserved

UEM\_TIME\_TAG, 128

rs1

UEM\_BM\_MESSAGE, 119

rs1\_d

UEM\_BM\_MESSAGE, 119

rs1\_pr

UEM\_BM\_MESSAGE, 119

rs2

UEM\_BM\_MESSAGE, 119

rs2\_d

UEM\_BM\_MESSAGE, 120

rs2\_pr

UEM\_BM\_MESSAGE, 119

secs

UEM\_TIME\_TAG, 128

size

UEM\_RAW\_BM\_MESSAGE, 124

UEM\_SEGMENT\_DESCR, 127

start\_time

UEM\_SEGMENT\_DESCR, 126

status

UEM\_RESP\_SEG, 125

uem.h, 130

UEM\_ABBUSY

Формирование и разбор командных и ответных слов, 116

UEM\_ABFAIL

Формирование и разбор командных и ответных слов, 116

UEM\_BAD\_SYNCHRO\_EEEEEI

Типы вносимых ошибок кодирования, 21

UEM\_BAD\_SYNCHRO\_EEEEIE

Типы вносимых ошибок кодирования, 21

UEM\_BAD\_SYNCHRO\_EEEIEE

Типы вносимых ошибок кодирования, 21

UEM\_BAD\_SYNCHRO\_EEIEEE

Типы вносимых ошибок кодирования, 21

UEM\_BAD\_SYNCHRO\_EIEEEE

Типы вносимых ошибок кодирования, 21

UEM\_BAD\_SYNCHRO\_IEEEEE

Типы вносимых ошибок кодирования, 21

UEM\_BAD\_SYNCHRO\_NONE

Типы вносимых ошибок кодирования, 21

UEM\_BAD\_SYNCHRO\_NONE2

Типы вносимых ошибок кодирования, 21

UEM\_BC

Действия с дескрипторами, 29

uem\_bc\_cseg\_create

Создание и настройка командных сегментов, 75

uem\_bc\_cseg\_format

Заполнение образа командного сегмента в ОЗУ ПЭВМ, 69

uem\_bc\_cseg\_format\_MODE

Заполнение образа командного сегмента в ОЗУ ПЭВМ, 70

uem\_bc\_cseg\_format\_RTRT

Заполнение образа командного сегмента в ОЗУ ПЭВМ, 70

uem\_bc\_cseg\_overlay

Создание и настройка командных сегментов, 79

uem\_bc\_gap\_create

Создание и настройка командных сегментов, 79

uem\_bc\_init

Установление и разрыв связи с устройством, 26

UEM\_BC\_RUNNING

Запуск и остановка, 105

uem\_bc\_send\_receive

Передача сообщений, 89

uem\_bc\_start

Запуск и остановка, 106

UEM\_BC\_START\_DEFAULT

Запуск и остановка, 105

UEM\_BC\_START\_NOW

Запуск и остановка, 105

UEM\_BC\_START\_WAITING

Запуск и остановка, 105

uem\_bc\_state\_ex

Запуск и остановка, 107

UEM\_BC\_STATE\_EX

Запуск и остановка, 105

uem\_bc\_stop

Запуск и остановка, 107

UEM\_BC\_STOP\_DEFAULT

Запуск и остановка, 105

UEM\_BC\_STOP\_NOW

Запуск и остановка, 105

UEM\_BC\_STOP\_ON\_FRAME

Запуск и остановка, 105

UEM\_BC\_STOPPED

Запуск и остановка, 105

UEM\_BC\_WAITING

Запуск и остановка, 105

UEM\_BCCA

Формирование и разбор командных и ответных слов, 116

UEM\_BCP

Действия с дескрипторами, 29

uem\_bcp\_append\_cseg

Создание и настройка кадров и программы КШ, 84

uem\_bcp\_append\_frame

Создание и настройка кадров и программы КШ, 84

uem\_bcp\_create

Создание и настройка кадров и программы КШ, 84

UEM\_BCP\_CUR\_SIZE

Создание и настройка кадров и программы КШ, 83

uem\_bcp\_desrtoy

Создание и настройка кадров и программы КШ, 86

uem\_bcp\_dimension

Создание и настройка кадров и программы КШ, 85

uem\_bcp\_discover\_cseg

Создание и настройка кадров и программы КШ, 85

uem\_bcp\_inspect\_frame

Создание и настройка кадров и программы КШ, 86

uem\_bcp\_install

Создание и настройка кадров и программы КШ, 86

UEM\_BCP\_MAX\_SIZE

Создание и настройка кадров и программы КШ, 83

UEM\_BCP\_NFRAMES

Создание и настройка кадров и программы КШ, 83

uem\_bcp\_replace\_cseg

Создание и настройка кадров и программы КШ, 85

uem\_bcp\_set\_standard\_gaps

Создание и настройка кадров и программы КШ, 87

UEM\_BCRT

Определения типов данных для КШ, ОУ, МШ, 63

UEM\_BCRTb

Определения типов данных для КШ, ОУ, МШ, 63

UEM\_BIPHASE\_POS\_MAX

Типы вносимых ошибок кодирования, 18

UEM\_BIPHASE\_POS\_MIN

Типы вносимых ошибок кодирования, 18

UEM\_BITCOUNT\_CHANGE\_MAX

Типы вносимых ошибок кодирования, 18

UEM\_BITCOUNT\_CHANGE\_MIN

Типы вносимых ошибок кодирования, 18

UEM\_BM

Действия с дескрипторами, 29

uem\_bm\_handler

Функции МШ, 102

uem\_bm\_init

Установление и разрыв связи с устройством, 27

uem\_bm\_install\_handler

Функции МШ, 103

UEM\_BM\_MESSAGE, 118

cs, 119

cs\_d, 119

errors, 120

gap, 120

lostp, 120

msg\_d, 119

overlay, 120

p\_pr, 120

raw, 120

rs1, 119

rs1\_d, 119

rs1\_pr, 119

rs2, 119

rs2\_d, 120

rs2\_pr, 119

uem\_bm\_queue\_count

Функции МШ, 103

uem\_bm\_receive

Функции МШ, 102

UEM\_BM\_WORD\_DEFAULT

Значения параметров, 54

UEM\_BM\_WORD\_MASK

Описание параметров, 41

UEM\_BM\_WORD\_MAX

Значения параметров, 54

UEM\_BM\_WORD\_MIN

Значения параметров, 54

UEM\_BM\_WORD\_PATTERN

Описание параметров, 41

UEM\_BOOL

Определения примитивных типов, 23

UEM\_BRCST\_DEFAULT

Значения параметров, 49

UEM\_BRCST\_DIS

Описание параметров, 36

UEM\_BRCST\_DISABLED

Значения параметров, 49

UEM\_BRCST\_ENABLED

Значения параметров, 49

UEM\_BRTF\_DEFAULT

Значения параметров, 52

UEM\_BRTF\_DIS

Описание параметров, 39

UEM\_BRTF\_DISABLED

Значения параметров, 52

UEM\_BRTF\_ENABLED

Значения параметров, 52

UEM\_BTMT\_DEFAULT

Значения параметров, 52

UEM\_BTMT\_DIS

Описание параметров, 39

UEM\_BTMT\_DISABLED

Значения параметров, 52

UEM\_BTMT\_ENABLED

Значения параметров, 52

UEM\_CH\_A

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_CH\_B

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_CHANNEL

Определения типов данных для КШ, ОУ, МШ, 66

uem\_close

Установление и разрыв связи с устройством, 27

UEM\_CMD\_SEG, 121

ch, 121

command1, 121

command2, 121

data, 122

format, 121

uem\_command\_word

Формирование и разбор командных и ответных слов, 116

uem\_command\_word\_parse

Формирование и разбор командных и ответных слов, 117

uem\_connect

Установление и разрыв связи с устройством, 26

UEM\_CSEG

Действия с дескрипторами, 29

uem\_cseg\_desrtoy

Создание и настройка командных сегментов, 80

uem\_cseg\_error\_get

Создание и настройка командных сегментов, 78

uem\_cseg\_error\_set

Создание и настройка командных сегментов, 77

UEM\_CSEG\_GAP

Создание и настройка командных сегментов, 75

UEM\_CSEG\_GAP\_ALT\_AB

Создание и настройка командных сегментов, 74

UEM\_CSEG\_GAP\_ALT\_BUS

Создание и настройка командных сегментов, 74

UEM\_CSEG\_GAP\_DEFAULT\_FLAGS

Создание и настройка командных сегментов, 74

UEM\_CSEG\_GAP\_DEFAULT\_TIMEOUT

Создание и настройка командных сегментов, 75

UEM\_CSEG\_GAP\_DEFAULT\_VALUE

Создание и настройка командных сегментов, 74

UEM\_CSEG\_GAP\_ESYNC

Создание и настройка командных сегментов, 73

UEM\_CSEG\_GAP\_FROM\_END

Создание и настройка командных сегментов, 73

UEM\_CSEG\_GAP\_FROM\_START

Создание и настройка командных сегментов, 73

uem\_cseg\_gap\_get

Создание и настройка командных сегментов, 76

UEM\_CSEG\_GAP\_MAX

Создание и настройка командных сегментов, 74

UEM\_CSEG\_GAP\_MIN

Создание и настройка командных сегментов, 74

uem\_cseg\_gap\_reset

Создание и настройка командных сегментов, 76

uem\_cseg\_gap\_set

Создание и настройка командных сегментов, 76

UEM\_CSEG\_GAP\_THIS\_AB

Создание и настройка командных сегментов, 74

UEM\_CSEG\_GAP\_THIS\_BUS

Создание и настройка командных сегментов, 74

UEM\_CSEG\_GAP\_TIMEOUT\_MAX

Создание и настройка командных сегментов, 75

UEM\_CSEG\_GAP\_TIMEOUT\_MIN

Создание и настройка командных сегментов, 74

UEM\_CSEG\_NORMAL

Создание и настройка командных сегментов, 75

UEM\_CSEG\_OVERLAY

Создание и настройка командных сегментов, 75

uem\_cseg\_read

Создание и настройка командных сегментов, 75

uem\_cseg\_sync\_get

Создание и настройка командных сегментов, 78

uem\_cseg\_sync\_set

Создание и настройка командных сегментов, 78

uem\_cseg\_type

Создание и настройка командных сегментов, 80

UEM\_CSEG\_TYPE

Создание и настройка командных сегментов, 75

uem\_cseg\_word\_gap\_get

Создание и настройка командных сегментов, 77

uem\_cseg\_word\_gap\_set

Создание и настройка командных сегментов, 77

UEM\_DATA, 123

data, 123

ndata, 123

UEM\_DB\_ACT

Описание параметров, 37

UEM\_DB\_ACTIVE

Значения параметров, 50

UEM\_DB\_INACTIVE

Значения параметров, 50

UEM\_DBCA

Формирование и разбор командных и ответных слов, 115

UEM\_DEVHANDLE

Определения примитивных типов, 23

UEM\_DWORD

Определения примитивных типов, 23

UEM\_ERR\_INJ\_DEFAULT

Значения параметров, 50

UEM\_ERR\_INJ\_DIS

Описание параметров, 36

UEM\_ERR\_INJ\_DISABLED

Значения параметров, 50

UEM\_ERR\_INJ\_ENABLED

Значения параметров, 50

UEM\_ERRF\_DT

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_ERRF\_ENC

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_ERRF\_ENCODING

Определения типов данных для КШ, ОУ, МШ, 64

UEM\_ERRF\_ENCODING2

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_ERRF\_ERROR

Определения типов данных для КШ, ОУ, МШ, 64

UEM\_ERRF\_EXTRA\_CWSW

Определения типов данных для КШ, ОУ, МШ, 65

UEM\_ERRF\_EXTRA\_DW

Определения типов данных для КШ, ОУ, МШ, 65

UEM\_ERRF\_FORMAT

Определения типов данных для КШ, ОУ, МШ, 64

UEM\_ERRF\_FORMAT\_MC

Определения типов данных для КШ, ОУ, МШ, 65

UEM\_ERRF\_GAPN

Определения типов данных для КШ, ОУ, МШ, 65

UEM\_ERRF\_INC\_MODE\_CODE

Определения типов данных для КШ, ОУ, МШ, 65

UEM\_ERRF\_INCORRECT\_RTN

Определения типов данных для КШ, ОУ, МШ, 65

UEM\_ERRF\_LESS\_BITS

Определения типов данных для КШ, ОУ, МШ, 65

UEM\_ERRF\_MINGAP

Определения типов данных для КШ, ОУ, МШ, 64

UEM\_ERRF\_MISSING\_CWSW

Определения типов данных для КШ, ОУ, МШ, 65

UEM\_ERRF\_MISSING\_DW

Определения типов данных для КШ, ОУ, МШ, 65

UEM\_ERRF\_MORE\_BITS

Определения типов данных для КШ, ОУ, МШ, 65

UEM\_ERRF\_NO\_RESPONSE

Определения типов данных для КШ, ОУ, МШ, 64

UEM\_ERRF\_PARITY

Определения типов данных для КШ, ОУ, МШ, 65

UEM\_ERRF\_RTRT\_FORMAT

Определения типов данных для КШ, ОУ, МШ, 65

UEM\_ERRF\_SYNC\_TYPE

Определения типов данных для КШ, ОУ, МШ, 64

UEM\_ERROR\_ADDRESS\_OUT\_OF\_RANGE

Коды завершения, 12

UEM\_ERROR\_BAD\_OVERLAY\_SOURCE

Коды завершения, 12

UEM\_ERROR\_BAD\_PARAM\_VALUE

Коды завершения, 10

UEM\_ERROR\_BAD\_PARAM\_VALUE\_1

Коды завершения, 10

UEM\_ERROR\_BAD\_PARAM\_VALUE\_10

Коды завершения, 11

UEM\_ERROR\_BAD\_PARAM\_VALUE\_2

Коды завершения, 11

UEM\_ERROR\_BAD\_PARAM\_VALUE\_3

Коды завершения, 11

UEM\_ERROR\_BAD\_PARAM\_VALUE\_4

Коды завершения, 11

UEM\_ERROR\_BAD\_PARAM\_VALUE\_5

Коды завершения, 11

UEM\_ERROR\_BAD\_PARAM\_VALUE\_6

Коды завершения, 11

UEM\_ERROR\_BAD\_PARAM\_VALUE\_7

Коды завершения, 11

UEM\_ERROR\_BAD\_PARAM\_VALUE\_8

Коды завершения, 11

UEM\_ERROR\_BAD\_PARAM\_VALUE\_9

Коды завершения, 11

UEM\_ERROR\_BAD\_TIMEOUT

Коды завершения, 12

UEM\_ERROR\_BCP\_NINST

Коды завершения, 12

UEM\_ERROR\_BM\_INTERNAL\_BUFFER\_OVERFLOW

Коды завершения, 13

UEM\_ERROR\_FLAGS

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_ERROR\_FORMAT\_DISABLED

Коды завершения, 12

UEM\_ERROR\_FORMAT\_X\_MCODE

Коды завершения, 12

UEM\_ERROR\_IN\_USE

Коды завершения, 13

UEM\_ERROR\_INC\_RESP

Коды завершения, 13

UEM\_ERROR\_INPOOL

Коды завершения, 12

UEM\_ERROR\_INV\_HANDLE

Коды завершения, 11

UEM\_ERROR\_INV\_HANDLE\_TYPE

Коды завершения, 11

UEM\_ERROR\_MAX\_SIZE\_EXCEED

Коды завершения, 13

uem\_error\_message

Служебные функции, 108

UEM\_ERROR\_NO\_FRAME\_APPEND

Коды завершения, 13

UEM\_ERROR\_NO\_FREE\_RAM

Коды завершения, 11

UEM\_ERROR\_NO\_HOST\_MEM

Коды завершения, 12

UEM\_ERROR\_NOT\_APPLICABLE

Коды завершения, 12

UEM\_ERROR\_NOT\_CONNECTED

Коды завершения, 12

UEM\_ERROR\_NUMBER\_OUT\_OF\_RANGE

Коды завершения, 12

UEM\_ERROR\_OFFSET

Коды завершения, 10

UEM\_ERROR\_PARAM\_DEFAULT

Типы вносимых ошибок кодирования, 19

UEM\_ERROR\_POS\_DEFAULT

Типы вносимых ошибок кодирования, 19

uem\_error\_query

Служебные функции, 109

UEM\_ERROR\_THREAD\_FAULT

Коды завершения, 13

UEM\_ERROR\_TOO\_MANY\_DATAWORDS

Коды завершения, 13

UEM\_ERROR\_TYPE

Типы вносимых ошибок кодирования, 19

UEM\_ERROR\_TYPE\_DEFAULT

Типы вносимых ошибок кодирования, 19

UEM\_ERROR\_WRONG\_LOCATION

Коды завершения, 12

UEM\_ERRT\_BAD\_BIPHASE\_NEG

Типы вносимых ошибок кодирования, 20

UEM\_ERRT\_BAD\_BIPHASE\_POS

Типы вносимых ошибок кодирования, 20

UEM\_ERRT\_BAD\_BIPHASE\_ZERO

Типы вносимых ошибок кодирования, 20

UEM\_ERRT\_BAD\_SYNCHRO

Типы вносимых ошибок кодирования, 19

UEM\_ERRT\_INV\_PARITY

Типы вносимых ошибок кодирования, 19

UEM\_ERRT\_NONE

Типы вносимых ошибок кодирования, 19

UEM\_ERRT\_SHIFT\_EDGE

Типы вносимых ошибок кодирования, 20

UEM\_ERRT\_WRONG\_BITCOUNT

Типы вносимых ошибок кодирования, 19

UEM\_F1

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_F10

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_F2

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_F3

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_F4

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_F5

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_F6

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_F7

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_F8

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_F9

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_FORMAT

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_FRAME\_ALLRPT

Создание и настройка кадров и программы КШ, 83

UEM\_FRAME\_CONT

Создание и настройка кадров и программы КШ, 83

UEM\_FRAME\_DEFAULT

Создание и настройка кадров и программы КШ, 83

UEM\_FRAME\_NONE

Создание и настройка кадров и программы КШ, 83

UEM\_FRAME\_REPEAT\_DEFAULT

Создание и настройка кадров и программы КШ, 82

UEM\_FRAME\_REPEAT\_MAX

Создание и настройка кадров и программы КШ, 82

UEM\_FRAME\_REPEAT\_MIN

Создание и настройка кадров и программы КШ, 82

UEM\_FRAME\_REPEAT\_UNLIM

Создание и настройка кадров и программы КШ, 82

UEM\_FRAME\_STOP

Создание и настройка кадров и программы КШ, 83

uem\_handle\_type

Действия с дескрипторами, 29

UEM\_HANDLE\_TYPE

Действия с дескрипторами, 29

uem\_init

Установление и разрыв связи с устройством, 25

UEM\_INVH

Действия с дескрипторами, 29

uem\_is\_running

Запуск и остановка, 106

UEM\_IST\_DEFAULT

Значения параметров, 54

UEM\_IST\_MAX

Значения параметров, 54

UEM\_IST\_MIN

Значения параметров, 54

UEM\_IST1

Описание параметров, 40

UEM\_IST2

Описание параметров, 40

uem\_layer\_handle

Действия с дескрипторами, 30

UEM\_LIB\_REV

Служебные функции, 108

UEM\_MC

Определения типов данных для КШ, ОУ, МШ, 63

UEM\_MC\_DEFAULT

Значения параметров, 51

UEM\_MC\_DIS

Описание параметров, 38

UEM\_MC\_DISABLED

Значения параметров, 51

UEM\_MC\_ENABLED

Значения параметров, 51

UEM\_MCb

Определения типов данных для КШ, ОУ, МШ, 64

UEM\_MCBCRT

Определения типов данных для КШ, ОУ, МШ, 63

UEM\_MCBCRTb

Определения типов данных для КШ, ОУ, МШ, 64

UEM\_MCODE\_ADBC

Формирование и разбор командных и ответных слов, 114

UEM\_MCODE\_BRTF

Формирование и разбор командных и ответных слов, 115

UEM\_MCODE\_BSELFTEST

Формирование и разбор командных и ответных слов, 114

UEM\_MCODE\_BTMT

Формирование и разбор командных и ответных слов, 114

UEM\_MCODE\_BTMT\_I

Формирование и разбор командных и ответных слов, 115

UEM\_MCODE\_RESETRT

Формирование и разбор командных и ответных слов, 115

UEM\_MCODE\_SYNCHRO

Формирование и разбор командных и ответных слов, 114

UEM\_MCODE\_SYNCHRO\_D

Формирование и разбор командных и ответных слов, 115

UEM\_MCODE\_TXBIT

Формирование и разбор командных и ответных слов, 115

UEM\_MCODE\_TXLCMD

Формирование и разбор командных и ответных слов, 115

UEM\_MCODE\_TXSTATUS

Формирование и разбор командных и ответных слов, 114

UEM\_MCODE\_TXVECT

Формирование и разбор командных и ответных слов, 115

UEM\_MCODE\_UBRTF

Формирование и разбор командных и ответных слов, 115

UEM\_MCODE\_UBTMT

Формирование и разбор командных и ответных слов, 115

UEM\_MCODE\_UBTMT\_I

Формирование и разбор командных и ответных слов, 115

UEM\_MCRTBC

Определения типов данных для КШ, ОУ, МШ, 63

UEM\_MIN\_T1

Параметры интервалов времени, 56

UEM\_MIN\_T1\_DEFAULT

Параметры интервалов времени, 56

UEM\_MIN\_T2

Параметры интервалов времени, 56

UEM\_MIN\_T2\_DEFAULT

Параметры интервалов времени, 56

UEM\_MODE\_0

Формирование и разбор командных и ответных слов, 114

UEM\_MODE\_31

Формирование и разбор командных и ответных слов, 114

UEM\_MSGERR

Формирование и разбор командных и ответных слов, 116

UEM\_NDATA\_BY\_CW

Определения типов данных для КШ, ОУ, МШ, 64

UEM\_NDATA\_MAX

Определения типов данных для КШ, ОУ, МШ, 64

UEM\_NDATA\_MIN

Определения типов данных для КШ, ОУ, МШ, 64

UEM\_OBJHANDLE

Определения примитивных типов, 23

uem\_param\_get

Параметры конфигурации УЭМ, 31

uem\_param\_set

Параметры конфигурации УЭМ, 32

UEM\_PARAMID

Определения примитивных типов, 23

uem\_parent\_dev

Действия с дескрипторами, 29

UEM\_RAW\_BM\_MESSAGE, 124

data, 124

size, 124

UEM\_RCVA\_DEFAULT

Значения параметров, 47

UEM\_RCVA\_DIS

Описание параметров, 34

UEM\_RCVA\_DISABLED

Значения параметров, 47

UEM\_RCVA\_ENABLED

Значения параметров, 47

UEM\_RCVB\_DEFAULT

Значения параметров, 48

UEM\_RCVB\_DIS

Описание параметров, 34

UEM\_RCVB\_DISABLED

Значения параметров, 47

UEM\_RCVB\_ENABLED

Значения параметров, 47

uem\_reset

Служебные функции, 109

UEM\_RESP

Действия с дескрипторами, 29

UEM\_RESP\_SEG, 125

data, 125

status, 125

uem\_response\_create

Функции ОУ, 95

uem\_response\_destroy

Функции ОУ, 101

uem\_response\_error\_get

Функции ОУ, 98

uem\_response\_error\_set

Функции ОУ, 97

uem\_response\_gap\_get

Функции ОУ, 96

uem\_response\_gap\_set

Функции ОУ, 96

uem\_response\_read

Функции ОУ, 96

uem\_response\_sync\_get

Функции ОУ, 98

uem\_response\_sync\_set

Функции ОУ, 98

uem\_response\_word\_gap\_get

Функции ОУ, 97

uem\_response\_word\_gap\_set

Функции ОУ, 97

uem\_revision\_query

Служебные функции, 109

UEM\_RFT\_DEFAULT

Значения параметров, 51

UEM\_RFT\_MAX

Значения параметров, 51

UEM\_RFT\_MIN

Значения параметров, 51

UEM\_RFT\_SINE

Значения параметров, 51

uem\_root\_dev

Действия с дескрипторами, 30

UEM\_RSV12

Формирование и разбор командных и ответных слов, 116

UEM\_RSV13

Формирование и разбор командных и ответных слов, 116

UEM\_RSV14

Формирование и разбор командных и ответных слов, 116

UEM\_RT

Действия с дескрипторами, 29

uem\_rt\_discover\_response

Функции ОУ, 100

uem\_rt\_discover\_response\_MODE

Функции ОУ, 100

uem\_rt\_init

Установление и разрыв связи с устройством, 27

uem\_rt\_install\_response

Функции ОУ, 99

uem\_rt\_install\_response\_MODE

Функции ОУ, 99

UEM\_RT\_RX

Формирование и разбор командных и ответных слов, 113

UEM\_RT\_TX

Формирование и разбор командных и ответных слов, 113

UEM\_RTADDR\_BRCST

Формирование и разбор командных и ответных слов, 113

UEM\_RTADDR\_MAX\_EXT

Формирование и разбор командных и ответных слов, 113

UEM\_RTADDR\_MIN

Формирование и разбор командных и ответных слов, 113

UEM\_RTADDR\_MАХ

Формирование и разбор командных и ответных слов, 113

UEM\_RTBC

Определения типов данных для КШ, ОУ, МШ, 63

UEM\_RTDES\_COM\_ILLEGAL

Функции ОУ, 92

UEM\_RTDES\_DBCA

Функции ОУ, 92

UEM\_RTDES\_DBCA\_BCSTART

Функции ОУ, 92

UEM\_RTDES\_DEFAULT

Функции ОУ, 95

UEM\_RTDES\_ILLEG\_MASK

Функции ОУ, 95

UEM\_RTDES\_LCMD\_DW

Функции ОУ, 92

UEM\_RTDES\_SW\_DIS

Функции ОУ, 91

UEM\_RTDES\_SWB\_SAV

Функции ОУ, 92

UEM\_RTDES\_WA

Функции ОУ, 94

UEM\_RTDES\_WA\_BRCST

Функции ОУ, 95

UEM\_RTDES\_WRONG\_CH

Функции ОУ, 94

UEM\_RTFAIL

Формирование и разбор командных и ответных слов, 115

UEM\_RTMO

Параметры интервалов времени, 56

UEM\_RTMO\_DEFAULT

Параметры интервалов времени, 56

UEM\_RTRT

Определения типов данных для КШ, ОУ, МШ, 63

UEM\_RTRTb

Определения типов данных для КШ, ОУ, МШ, 64

UEM\_SADDR\_CONV\_LOOPBACK

Формирование и разбор командных и ответных слов, 114

UEM\_SADDR\_MAX

Формирование и разбор командных и ответных слов, 114

UEM\_SADDR\_MAX\_EXT

Формирование и разбор командных и ответных слов, 114

UEM\_SADDR\_MIN

Формирование и разбор командных и ответных слов, 114

UEM\_SADDR\_MIN\_EXT

Формирование и разбор командных и ответных слов, 114

UEM\_SEGMENT\_DESCR, 126

dwoffset, 127

end\_time, 126

endoffset, 127

errors, 126

offset, 126

size, 127

start\_time, 126

UEM\_SEL\_UNBASE\_INT

Действия с дескрипторами, 28

UEM\_SEL\_UNMBASE

Действия с дескрипторами, 28

UEM\_SEL\_UNMUEM

Действия с дескрипторами, 28

uem\_self\_test

Служебные функции, 110

UEM\_SERVRQ

Формирование и разбор командных и ответных слов, 116

UEM\_SET

Значения параметров, 50

UEM\_SHIFT\_LENGTH\_MAX

Типы вносимых ошибок кодирования, 19

UEM\_SHIFT\_LENGTH\_MIN

Типы вносимых ошибок кодирования, 19

UEM\_SHIFT\_POS\_MAX

Типы вносимых ошибок кодирования, 19

UEM\_SHIFT\_POS\_MIN

Типы вносимых ошибок кодирования, 18

uem\_start

Запуск и остановка, 106

uem\_status\_word

Формирование и разбор командных и ответных слов, 117

uem\_status\_word\_parse

Формирование и разбор командных и ответных слов, 117

uem\_stop

Запуск и остановка, 106

UEM\_SYNC

Определения типов данных для КШ, ОУ, МШ, 67

UEM\_SYNC\_C

Определения типов данных для КШ, ОУ, МШ, 67

UEM\_SYNC\_D

Определения типов данных для КШ, ОУ, МШ, 67

UEM\_SYNC\_IN\_1\_DEFAULT

Значения параметров, 48

UEM\_SYNC\_IN\_1\_DISABLED

Значения параметров, 48

UEM\_SYNC\_IN\_1\_ENA

Описание параметров, 35

UEM\_SYNC\_IN\_1\_ENABLED

Значения параметров, 48

UEM\_SYNC\_IN\_1\_INTGEN

Описание параметров, 36

UEM\_SYNC\_IN\_1\_INTGEN\_DEFAULT

Значения параметров, 49

UEM\_SYNC\_IN\_1\_INTGEN\_DISABLED

Значения параметров, 49

UEM\_SYNC\_IN\_1\_INTGEN\_ENABLED

Значения параметров, 49

UEM\_SYNC\_IN\_1\_SET

Описание параметров, 37

UEM\_SYNC\_IN\_2\_DEFAULT

Значения параметров, 48

UEM\_SYNC\_IN\_2\_DISABLED

Значения параметров, 48

UEM\_SYNC\_IN\_2\_ENA

Описание параметров, 35

UEM\_SYNC\_IN\_2\_ENABLED

Значения параметров, 48

UEM\_SYNC\_IN\_2\_INTGEN

Описание параметров, 36

UEM\_SYNC\_IN\_2\_INTGEN\_DEFAULT

Значения параметров, 50

UEM\_SYNC\_IN\_2\_INTGEN\_DISABLED

Значения параметров, 50

UEM\_SYNC\_IN\_2\_INTGEN\_ENABLED

Значения параметров, 50

UEM\_SYNC\_IN\_2\_SET

Описание параметров, 37

UEM\_SYNC\_OUT\_1\_DEFAULT

Значения параметров, 49

UEM\_SYNC\_OUT\_1\_DISABLED

Значения параметров, 48

UEM\_SYNC\_OUT\_1\_ENA

Описание параметров, 35

UEM\_SYNC\_OUT\_1\_ENABLED

Значения параметров, 48

UEM\_SYNC\_OUT\_1\_SET

Описание параметров, 37

UEM\_SYNC\_OUT\_2\_DEFAULT

Значения параметров, 49

UEM\_SYNC\_OUT\_2\_DISABLED

Значения параметров, 49

UEM\_SYNC\_OUT\_2\_ENA

Описание параметров, 35

UEM\_SYNC\_OUT\_2\_ENABLED

Значения параметров, 49

UEM\_SYNC\_OUT\_2\_SET

Описание параметров, 37

UEM\_SYNC1\_ACH

Значения параметров, 54

UEM\_SYNC1\_C\_D\_

Описание параметров, 39

UEM\_SYNC1\_C\_D\_DEFAULT

Значения параметров, 53

UEM\_SYNC1\_CH

Описание параметров, 40

UEM\_SYNC1\_CH\_A

Значения параметров, 53

UEM\_SYNC1\_CH\_B

Значения параметров, 54

UEM\_SYNC1\_CH\_DEFAULT

Значения параметров, 54

UEM\_SYNC1\_ERR

Описание параметров, 39

UEM\_SYNC1\_ERR\_DEFAULT

Значения параметров, 53

UEM\_SYNC1\_GAPB

Описание параметров, 40

UEM\_SYNC1\_GAPB\_DEFAULT

Значения параметров, 53

UEM\_SYNC1\_ON\_COMMAND

Значения параметров, 53

UEM\_SYNC1\_ON\_DATA

Значения параметров, 53

UEM\_SYNC1\_ON\_ERROR

Значения параметров, 53

UEM\_SYNC1\_ON\_GAPB

Значения параметров, 53

UEM\_SYNC1\_ON\_NO\_ERROR

Значения параметров, 53

UEM\_SYNC1\_ON\_NO\_GAPB

Значения параметров, 53

UEM\_SYNC2\_VRTA

Описание параметров, 39

UEM\_SYNC2\_VRTA\_DEFAULT

Значения параметров, 52

UEM\_SYNC2\_VRTA\_MAX

Значения параметров, 52

UEM\_SYNC2\_VRTA\_MIN

Значения параметров, 52

UEM\_SYNCHRO\_ERROR\_POS

Типы вносимых ошибок кодирования, 20

UEM\_TIME\_PARAM

Параметры интервалов времени, 56

UEM\_TIME\_TAG, 128

b, 129

days, 128

i, 129

quartas, 128

reserved, 128

secs, 128

uem\_time\_tag\_get

Встроенный счетчик времени, 59

UEM\_TIME\_TAG\_LIN

Встроенный счетчик времени, 59

uem\_time\_tag\_reset

Встроенный счетчик времени, 59

uem\_time\_tag\_set

Встроенный счетчик времени, 59

uem\_time\_tag\_to\_linear

Встроенный счетчик времени, 60

uem\_time\_tag\_to\_struct

Встроенный счетчик времени, 60

uem\_timing\_get

Параметры интервалов времени, 57

uem\_timing\_set

Параметры интервалов времени, 56

UEM\_TMT\_RES

Описание параметров, 36

UEM\_TMTA\_DEFAULT

Значения параметров, 47

UEM\_TMTA\_DIS

Описание параметров, 34

UEM\_TMTA\_DISABLED

Значения параметров, 46

UEM\_TMTA\_ENABLED

Значения параметров, 47

UEM\_TMTB\_DEFAULT

Значения параметров, 47

UEM\_TMTB\_DIS

Описание параметров, 34

UEM\_TMTB\_DISABLED

Значения параметров, 47

UEM\_TMTB\_ENABLED

Значения параметров, 47

UEM\_TXA\_RFT

Описание параметров, 38

UEM\_TXA\_VPP

Описание параметров, 38

UEM\_TXB\_RFT

Описание параметров, 38

UEM\_TXB\_VPP

Описание параметров, 38

UEM\_UEM

Действия с дескрипторами, 29

UEM\_UNF

Определения типов данных для КШ, ОУ, МШ, 66

UEM\_VPP\_DEFAULT

Значения параметров, 51

UEM\_VPP\_MAX

Значения параметров, 51

UEM\_VPP\_MIN

Значения параметров, 51

UEM\_WARN\_JUST\_IN\_STATE

Коды завершения, 13

UEM\_WARN\_NO\_NEXT\_MESSAGE

Коды завершения, 13

UEM\_WARN\_OFFSET

Коды завершения, 10

UEM\_WORD

Определения примитивных типов, 24

UEM\_ZERO

Формирование и разбор командных и ответных слов, 116

Адресная строка, 14

Вводные и дополнительные сведения, 8

Виртуальные устройства, 14

Внесение ошибок состава сообщения, 70

Встроенный счетчик времени, 58

uem\_time\_tag\_get, 59

UEM\_TIME\_TAG\_LIN, 59

uem\_time\_tag\_reset, 59

uem\_time\_tag\_set, 59

uem\_time\_tag\_to\_linear, 60

uem\_time\_tag\_to\_struct, 60

Действия с дескрипторами, 28

UEM\_BC, 29

UEM\_BCP, 29

UEM\_BM, 29

UEM\_CSEG, 29

uem\_handle\_type, 29

UEM\_HANDLE\_TYPE, 29

UEM\_INVH, 29

uem\_layer\_handle, 30

uem\_parent\_dev, 29

UEM\_RESP, 29

uem\_root\_dev, 30

UEM\_RT, 29

UEM\_SEL\_UNBASE\_INT, 28

UEM\_SEL\_UNMBASE, 28

UEM\_SEL\_UNMUEM, 28

UEM\_UEM, 29

Заполнение образа командного сегмента в ОЗУ ПЭВМ, 68

uem\_bc\_cseg\_format, 69

uem\_bc\_cseg\_format\_MODE, 70

uem\_bc\_cseg\_format\_RTRT, 70

Запуск и остановка, 104

UEM\_BC\_RUNNING, 105

uem\_bc\_start, 106

UEM\_BC\_START\_DEFAULT, 105

UEM\_BC\_START\_NOW, 105

UEM\_BC\_START\_WAITING, 105

uem\_bc\_state\_ex, 107

UEM\_BC\_STATE\_EX, 105

uem\_bc\_stop, 107

UEM\_BC\_STOP\_DEFAULT, 105

UEM\_BC\_STOP\_NOW, 105

UEM\_BC\_STOP\_ON\_FRAME, 105

UEM\_BC\_STOPPED, 105

UEM\_BC\_WAITING, 105

uem\_is\_running, 106

uem\_start, 106

uem\_stop, 106

Запуск и остановка КШ, 87

Значения параметров, 41

UEM\_BM\_WORD\_DEFAULT, 54

UEM\_BM\_WORD\_MAX, 54

UEM\_BM\_WORD\_MIN, 54

UEM\_BRCST\_DEFAULT, 49

UEM\_BRCST\_DISABLED, 49

UEM\_BRCST\_ENABLED, 49

UEM\_BRTF\_DEFAULT, 52

UEM\_BRTF\_DISABLED, 52

UEM\_BRTF\_ENABLED, 52

UEM\_BTMT\_DEFAULT, 52

UEM\_BTMT\_DISABLED, 52

UEM\_BTMT\_ENABLED, 52

UEM\_DB\_ACTIVE, 50

UEM\_DB\_INACTIVE, 50

UEM\_ERR\_INJ\_DEFAULT, 50

UEM\_ERR\_INJ\_DISABLED, 50

UEM\_ERR\_INJ\_ENABLED, 50

UEM\_IST\_DEFAULT, 54

UEM\_IST\_MAX, 54

UEM\_IST\_MIN, 54

UEM\_MC\_DEFAULT, 51

UEM\_MC\_DISABLED, 51

UEM\_MC\_ENABLED, 51

UEM\_RCVA\_DEFAULT, 47

UEM\_RCVA\_DISABLED, 47

UEM\_RCVA\_ENABLED, 47

UEM\_RCVB\_DEFAULT, 48

UEM\_RCVB\_DISABLED, 47

UEM\_RCVB\_ENABLED, 47

UEM\_RFT\_DEFAULT, 51

UEM\_RFT\_MAX, 51

UEM\_RFT\_MIN, 51

UEM\_RFT\_SINE, 51

UEM\_SET, 50

UEM\_SYNC\_IN\_1\_DEFAULT, 48

UEM\_SYNC\_IN\_1\_DISABLED, 48

UEM\_SYNC\_IN\_1\_ENABLED, 48

UEM\_SYNC\_IN\_1\_INTGEN\_DEFAULT, 49

UEM\_SYNC\_IN\_1\_INTGEN\_DISABLED, 49

UEM\_SYNC\_IN\_1\_INTGEN\_ENABLED, 49

UEM\_SYNC\_IN\_2\_DEFAULT, 48

UEM\_SYNC\_IN\_2\_DISABLED, 48

UEM\_SYNC\_IN\_2\_ENABLED, 48

UEM\_SYNC\_IN\_2\_INTGEN\_DEFAULT, 50

UEM\_SYNC\_IN\_2\_INTGEN\_DISABLED, 50

UEM\_SYNC\_IN\_2\_INTGEN\_ENABLED, 50

UEM\_SYNC\_OUT\_1\_DEFAULT, 49

UEM\_SYNC\_OUT\_1\_DISABLED, 48

UEM\_SYNC\_OUT\_1\_ENABLED, 48

UEM\_SYNC\_OUT\_2\_DEFAULT, 49

UEM\_SYNC\_OUT\_2\_DISABLED, 49

UEM\_SYNC\_OUT\_2\_ENABLED, 49

UEM\_SYNC1\_ACH, 54

UEM\_SYNC1\_C\_D\_DEFAULT, 53

UEM\_SYNC1\_CH\_A, 53

UEM\_SYNC1\_CH\_B, 54

UEM\_SYNC1\_CH\_DEFAULT, 54

UEM\_SYNC1\_ERR\_DEFAULT, 53

UEM\_SYNC1\_GAPB\_DEFAULT, 53

UEM\_SYNC1\_ON\_COMMAND, 53

UEM\_SYNC1\_ON\_DATA, 53

UEM\_SYNC1\_ON\_ERROR, 53

UEM\_SYNC1\_ON\_GAPB, 53

UEM\_SYNC1\_ON\_NO\_ERROR, 53

UEM\_SYNC1\_ON\_NO\_GAPB, 53

UEM\_SYNC2\_VRTA\_DEFAULT, 52

UEM\_SYNC2\_VRTA\_MAX, 52

UEM\_SYNC2\_VRTA\_MIN, 52

UEM\_TMTA\_DEFAULT, 47

UEM\_TMTA\_DISABLED, 46

UEM\_TMTA\_ENABLED, 47

UEM\_TMTB\_DEFAULT, 47

UEM\_TMTB\_DISABLED, 47

UEM\_TMTB\_ENABLED, 47

UEM\_VPP\_DEFAULT, 51

UEM\_VPP\_MAX, 51

UEM\_VPP\_MIN, 51

Коды завершения, 8

UEM\_ERROR\_ADDRESS\_OUT\_OF\_RANGE, 12

UEM\_ERROR\_BAD\_OVERLAY\_SOURCE, 12

UEM\_ERROR\_BAD\_PARAM\_VALUE, 10

UEM\_ERROR\_BAD\_PARAM\_VALUE\_1, 10

UEM\_ERROR\_BAD\_PARAM\_VALUE\_10, 11

UEM\_ERROR\_BAD\_PARAM\_VALUE\_2, 11

UEM\_ERROR\_BAD\_PARAM\_VALUE\_3, 11

UEM\_ERROR\_BAD\_PARAM\_VALUE\_4, 11

UEM\_ERROR\_BAD\_PARAM\_VALUE\_5, 11

UEM\_ERROR\_BAD\_PARAM\_VALUE\_6, 11

UEM\_ERROR\_BAD\_PARAM\_VALUE\_7, 11

UEM\_ERROR\_BAD\_PARAM\_VALUE\_8, 11

UEM\_ERROR\_BAD\_PARAM\_VALUE\_9, 11

UEM\_ERROR\_BAD\_TIMEOUT, 12

UEM\_ERROR\_BCP\_NINST, 12

UEM\_ERROR\_BM\_INTERNAL\_BUFFER\_OVERFLOW, 13

UEM\_ERROR\_FORMAT\_DISABLED, 12

UEM\_ERROR\_FORMAT\_X\_MCODE, 12

UEM\_ERROR\_IN\_USE, 13

UEM\_ERROR\_INC\_RESP, 13

UEM\_ERROR\_INPOOL, 12

UEM\_ERROR\_INV\_HANDLE, 11

UEM\_ERROR\_INV\_HANDLE\_TYPE, 11

UEM\_ERROR\_MAX\_SIZE\_EXCEED, 13

UEM\_ERROR\_NO\_FRAME\_APPEND, 13

UEM\_ERROR\_NO\_FREE\_RAM, 11

UEM\_ERROR\_NO\_HOST\_MEM, 12

UEM\_ERROR\_NOT\_APPLICABLE, 12

UEM\_ERROR\_NOT\_CONNECTED, 12

UEM\_ERROR\_NUMBER\_OUT\_OF\_RANGE, 12

UEM\_ERROR\_OFFSET, 10

UEM\_ERROR\_THREAD\_FAULT, 13

UEM\_ERROR\_TOO\_MANY\_DATAWORDS, 13

UEM\_ERROR\_WRONG\_LOCATION, 12

UEM\_WARN\_JUST\_IN\_STATE, 13

UEM\_WARN\_NO\_NEXT\_MESSAGE, 13

UEM\_WARN\_OFFSET, 10

Командные и ответные сегменты, 14

Описание параметров, 32

UEM\_BM\_WORD\_MASK, 41

UEM\_BM\_WORD\_PATTERN, 41

UEM\_BRCST\_DIS, 36

UEM\_BRTF\_DIS, 39

UEM\_BTMT\_DIS, 39

UEM\_DB\_ACT, 37

UEM\_ERR\_INJ\_DIS, 36

UEM\_IST1, 40

UEM\_IST2, 40

UEM\_MC\_DIS, 38

UEM\_RCVA\_DIS, 34

UEM\_RCVB\_DIS, 34

UEM\_SYNC\_IN\_1\_ENA, 35

UEM\_SYNC\_IN\_1\_INTGEN, 36

UEM\_SYNC\_IN\_1\_SET, 37

UEM\_SYNC\_IN\_2\_ENA, 35

UEM\_SYNC\_IN\_2\_INTGEN, 36

UEM\_SYNC\_IN\_2\_SET, 37

UEM\_SYNC\_OUT\_1\_ENA, 35

UEM\_SYNC\_OUT\_1\_SET, 37

UEM\_SYNC\_OUT\_2\_ENA, 35

UEM\_SYNC\_OUT\_2\_SET, 37

UEM\_SYNC1\_C\_D\_, 39

UEM\_SYNC1\_CH, 40

UEM\_SYNC1\_ERR, 39

UEM\_SYNC1\_GAPB, 40

UEM\_SYNC2\_VRTA, 39

UEM\_TMT\_RES, 36

UEM\_TMTA\_DIS, 34

UEM\_TMTB\_DIS, 34

UEM\_TXA\_RFT, 38

UEM\_TXA\_VPP, 38

UEM\_TXB\_RFT, 38

UEM\_TXB\_VPP, 38

Определения примитивных типов, 23

UEM\_BOOL, 23

UEM\_DEVHANDLE, 23

UEM\_DWORD, 23

UEM\_OBJHANDLE, 23

UEM\_PARAMID, 23

UEM\_WORD, 24

Определения типов данных для КШ, ОУ, МШ, 61

UEM\_BCRT, 63

UEM\_BCRTb, 63

UEM\_CH\_A, 66

UEM\_CH\_B, 66

UEM\_CHANNEL, 66

UEM\_ERRF\_DT, 66

UEM\_ERRF\_ENC, 66

UEM\_ERRF\_ENCODING, 64

UEM\_ERRF\_ENCODING2, 66

UEM\_ERRF\_ERROR, 64

UEM\_ERRF\_EXTRA\_CWSW, 65

UEM\_ERRF\_EXTRA\_DW, 65

UEM\_ERRF\_FORMAT, 64

UEM\_ERRF\_FORMAT\_MC, 65

UEM\_ERRF\_GAPN, 65

UEM\_ERRF\_INC\_MODE\_CODE, 65

UEM\_ERRF\_INCORRECT\_RTN, 65

UEM\_ERRF\_LESS\_BITS, 65

UEM\_ERRF\_MINGAP, 64

UEM\_ERRF\_MISSING\_CWSW, 65

UEM\_ERRF\_MISSING\_DW, 65

UEM\_ERRF\_MORE\_BITS, 65

UEM\_ERRF\_NO\_RESPONSE, 64

UEM\_ERRF\_PARITY, 65

UEM\_ERRF\_RTRT\_FORMAT, 65

UEM\_ERRF\_SYNC\_TYPE, 64

UEM\_ERROR\_FLAGS, 66

UEM\_F1, 66

UEM\_F10, 66

UEM\_F2, 66

UEM\_F3, 66

UEM\_F4, 66

UEM\_F5, 66

UEM\_F6, 66

UEM\_F7, 66

UEM\_F8, 66

UEM\_F9, 66

UEM\_FORMAT, 66

UEM\_MC, 63

UEM\_MCb, 64

UEM\_MCBCRT, 63

UEM\_MCBCRTb, 64

UEM\_MCRTBC, 63

UEM\_NDATA\_BY\_CW, 64

UEM\_NDATA\_MAX, 64

UEM\_NDATA\_MIN, 64

UEM\_RTBC, 63

UEM\_RTRT, 63

UEM\_RTRTb, 64

UEM\_SYNC, 67

UEM\_SYNC\_C, 67

UEM\_SYNC\_D, 67

UEM\_UNF, 66

Параметры интервалов времени, 55

UEM\_MIN\_T1, 56

UEM\_MIN\_T1\_DEFAULT, 56

UEM\_MIN\_T2, 56

UEM\_MIN\_T2\_DEFAULT, 56

UEM\_RTMO, 56

UEM\_RTMO\_DEFAULT, 56

UEM\_TIME\_PARAM, 56

uem\_timing\_get, 57

uem\_timing\_set, 56

Параметры конфигурации УЭМ, 31

uem\_param\_get, 31

uem\_param\_set, 32

Параметры передатчиков и характеристики выходных сигналов, 15

Передача сообщений, 88

uem\_bc\_send\_receive, 89

Порядок действий при установлении связи с устройством, 13

Служебные функции, 108

uem\_error\_message, 108

uem\_error\_query, 109

UEM\_LIB\_REV, 108

uem\_reset, 109

uem\_revision\_query, 109

uem\_self\_test, 110

Создание и настройка кадров и программы КШ, 81

uem\_bcp\_append\_cseg, 84

uem\_bcp\_append\_frame, 84

uem\_bcp\_create, 84

UEM\_BCP\_CUR\_SIZE, 83

uem\_bcp\_desrtoy, 86

uem\_bcp\_dimension, 85

uem\_bcp\_discover\_cseg, 85

uem\_bcp\_inspect\_frame, 86

uem\_bcp\_install, 86

UEM\_BCP\_MAX\_SIZE, 83

UEM\_BCP\_NFRAMES, 83

uem\_bcp\_replace\_cseg, 85

uem\_bcp\_set\_standard\_gaps, 87

UEM\_FRAME\_ALLRPT, 83

UEM\_FRAME\_CONT, 83

UEM\_FRAME\_DEFAULT, 83

UEM\_FRAME\_NONE, 83

UEM\_FRAME\_REPEAT\_DEFAULT, 82

UEM\_FRAME\_REPEAT\_MAX, 82

UEM\_FRAME\_REPEAT\_MIN, 82

UEM\_FRAME\_REPEAT\_UNLIM, 82

UEM\_FRAME\_STOP, 83

Создание и настройка командных сегментов, 71

uem\_bc\_cseg\_create, 75

uem\_bc\_cseg\_overlay, 79

uem\_bc\_gap\_create, 79

uem\_cseg\_desrtoy, 80

uem\_cseg\_error\_get, 78

uem\_cseg\_error\_set, 77

UEM\_CSEG\_GAP, 75

UEM\_CSEG\_GAP\_ALT\_AB, 74

UEM\_CSEG\_GAP\_ALT\_BUS, 74

UEM\_CSEG\_GAP\_DEFAULT\_FLAGS, 74

UEM\_CSEG\_GAP\_DEFAULT\_TIMEOUT, 75

UEM\_CSEG\_GAP\_DEFAULT\_VALUE, 74

UEM\_CSEG\_GAP\_ESYNC, 73

UEM\_CSEG\_GAP\_FROM\_END, 73

UEM\_CSEG\_GAP\_FROM\_START, 73

uem\_cseg\_gap\_get, 76

UEM\_CSEG\_GAP\_MAX, 74

UEM\_CSEG\_GAP\_MIN, 74

uem\_cseg\_gap\_reset, 76

uem\_cseg\_gap\_set, 76

UEM\_CSEG\_GAP\_THIS\_AB, 74

UEM\_CSEG\_GAP\_THIS\_BUS, 74

UEM\_CSEG\_GAP\_TIMEOUT\_MAX, 75

UEM\_CSEG\_GAP\_TIMEOUT\_MIN, 74

UEM\_CSEG\_NORMAL, 75

UEM\_CSEG\_OVERLAY, 75

uem\_cseg\_read, 75

uem\_cseg\_sync\_get, 78

uem\_cseg\_sync\_set, 78

uem\_cseg\_type, 80

UEM\_CSEG\_TYPE, 75

uem\_cseg\_word\_gap\_get, 77

uem\_cseg\_word\_gap\_set, 77

Типы вносимых ошибок кодирования, 17

UEM\_BAD\_SYNCHRO\_EEEEEI, 21

UEM\_BAD\_SYNCHRO\_EEEEIE, 21

UEM\_BAD\_SYNCHRO\_EEEIEE, 21

UEM\_BAD\_SYNCHRO\_EEIEEE, 21

UEM\_BAD\_SYNCHRO\_EIEEEE, 21

UEM\_BAD\_SYNCHRO\_IEEEEE, 21

UEM\_BAD\_SYNCHRO\_NONE, 21

UEM\_BAD\_SYNCHRO\_NONE2, 21

UEM\_BIPHASE\_POS\_MAX, 18

UEM\_BIPHASE\_POS\_MIN, 18

UEM\_BITCOUNT\_CHANGE\_MAX, 18

UEM\_BITCOUNT\_CHANGE\_MIN, 18

UEM\_ERROR\_PARAM\_DEFAULT, 19

UEM\_ERROR\_POS\_DEFAULT, 19

UEM\_ERROR\_TYPE, 19

UEM\_ERROR\_TYPE\_DEFAULT, 19

UEM\_ERRT\_BAD\_BIPHASE\_NEG, 20

UEM\_ERRT\_BAD\_BIPHASE\_POS, 20

UEM\_ERRT\_BAD\_BIPHASE\_ZERO, 20

UEM\_ERRT\_BAD\_SYNCHRO, 19

UEM\_ERRT\_INV\_PARITY, 19

UEM\_ERRT\_NONE, 19

UEM\_ERRT\_SHIFT\_EDGE, 20

UEM\_ERRT\_WRONG\_BITCOUNT, 19

UEM\_SHIFT\_LENGTH\_MAX, 19

UEM\_SHIFT\_LENGTH\_MIN, 19

UEM\_SHIFT\_POS\_MAX, 19

UEM\_SHIFT\_POS\_MIN, 18

UEM\_SYNCHRO\_ERROR\_POS, 20

Управление синхронизацией, 21

Установление и разрыв связи с устройством, 25

uem\_bc\_init, 26

uem\_bm\_init, 27

uem\_close, 27

uem\_connect, 26

uem\_init, 25

uem\_rt\_init, 27

Формирование и разбор командных и ответных слов, 111

UEM\_ABBUSY, 116

UEM\_ABFAIL, 116

UEM\_BCCA, 116

uem\_command\_word, 116

uem\_command\_word\_parse, 117

UEM\_DBCA, 115

UEM\_MCODE\_ADBC, 114

UEM\_MCODE\_BRTF, 115

UEM\_MCODE\_BSELFTEST, 114

UEM\_MCODE\_BTMT, 114

UEM\_MCODE\_BTMT\_I, 115

UEM\_MCODE\_RESETRT, 115

UEM\_MCODE\_SYNCHRO, 114

UEM\_MCODE\_SYNCHRO\_D, 115

UEM\_MCODE\_TXBIT, 115

UEM\_MCODE\_TXLCMD, 115

UEM\_MCODE\_TXSTATUS, 114

UEM\_MCODE\_TXVECT, 115

UEM\_MCODE\_UBRTF, 115

UEM\_MCODE\_UBTMT, 115

UEM\_MCODE\_UBTMT\_I, 115

UEM\_MODE\_0, 114

UEM\_MODE\_31, 114

UEM\_MSGERR, 116

UEM\_RSV12, 116

UEM\_RSV13, 116

UEM\_RSV14, 116

UEM\_RT\_RX, 113

UEM\_RT\_TX, 113

UEM\_RTADDR\_BRCST, 113

UEM\_RTADDR\_MAX\_EXT, 113

UEM\_RTADDR\_MIN, 113

UEM\_RTADDR\_MАХ, 113

UEM\_RTFAIL, 115

UEM\_SADDR\_CONV\_LOOPBACK, 114

UEM\_SADDR\_MAX, 114

UEM\_SADDR\_MAX\_EXT, 114

UEM\_SADDR\_MIN, 114

UEM\_SADDR\_MIN\_EXT, 114

UEM\_SERVRQ, 116

uem\_status\_word, 117

uem\_status\_word\_parse, 117

UEM\_ZERO, 116

Функции КШ, 68

Функции МШ, 102

uem\_bm\_handler, 102

uem\_bm\_install\_handler, 103

uem\_bm\_queue\_count, 103

uem\_bm\_receive, 102

Функции ОУ, 90

uem\_response\_create, 95

uem\_response\_destroy, 101

uem\_response\_error\_get, 98

uem\_response\_error\_set, 97

uem\_response\_gap\_get, 96

uem\_response\_gap\_set, 96

uem\_response\_read, 96

uem\_response\_sync\_get, 98

uem\_response\_sync\_set, 98

uem\_response\_word\_gap\_get, 97

uem\_response\_word\_gap\_set, 97

uem\_rt\_discover\_response, 100

uem\_rt\_discover\_response\_MODE, 100

uem\_rt\_install\_response, 99

uem\_rt\_install\_response\_MODE, 99

UEM\_RTDES\_COM\_ILLEGAL, 92

UEM\_RTDES\_DBCA, 92

UEM\_RTDES\_DBCA\_BCSTART, 92

UEM\_RTDES\_DEFAULT, 95

UEM\_RTDES\_ILLEG\_MASK, 95

UEM\_RTDES\_LCMD\_DW, 92

UEM\_RTDES\_SW\_DIS, 91

UEM\_RTDES\_SWB\_SAV, 92

UEM\_RTDES\_WA, 94

UEM\_RTDES\_WA\_BRCST, 95

UEM\_RTDES\_WRONG\_CH, 94

# ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

|  |  |
| --- | --- |
| АЗ | Абонент занят |
| ВСК | Встроенный самоконтроль |
| ЗО | Запрос обслуживания |
| КС | Командное слово |
| КУ | Команда управления |
| КШ | Контроллер шины |
| КШОУ | Тип сообщения – от КШ к ОУ |
| МЗР | Младший значащий разряд |
| МК | Мультиплексный канал |
| МКПД | Мультиплексный канал передачи данных |
| МШ | Монитор шины |
| НА | Неисправность абонента |
| НОУ | Неисправность ОУ |
| НП | Непосредственное подключение |
| НС | Нормальное состояние |
| НФ | Неформатное сообщение |
| ОЗУ | Оперативное запоминающее устройство |
| ОО | Отсутствие ответа |
| ОС | Ответное слово |
| ОУ | Оконечное устройство |
| ОУКШ | Тип сообщения – от ОУ к КШ |
| ОУОУ | Тип сообщения – от ОУ к ОУ |
| ОШС | Ошибка в сообщении |
| ПА | Подадрес |
| ПГК | Принята групповая команда |
| ПО | Программное обеспечение |
| ПУИ | Принято управление интерфейсом |
| ПЭВМ | Персональная электронная вычислительная машина |
| РЕЗ | Резервные признаки ответного слова |
| СД | Слово данных |
| ТП | Трансформаторное подключение |
| УЭМ | Универсальный электронный модуль |

# ПЕРЕЧЕНЬ ССЫЛОЧНЫХ ДОКУМЕНТОВ

1. ГОСТ Р 52070-2003. Интерфейс магистральный последовательный системы электронных модулей. Общие требования.

2. ЮФКВ.469555.555РЭ. Универсальный электронный модуль УЭМ-МК. Руководство по эксплуатации.

3. ЮФКВ.469555.731РЭ Модуль МВ98.03. Руководство по эксплуатации.

4. ФТКС.76902-01 32 01. ДРАЙВЕР НМ. Руководство системного программиста.

5. ЮФКВ. 10149-01 32 01. Универсальный электронный модуль УЭМ-МК. Программное обеспечение системное. Руководство системного программиста.

6. ЮФКВ.10175-01 32 01. Модуль МВ98.03. Программное обеспечение системное. Руководство системного программиста.

| **Лист регистрации изменений** | | | | | | | | | |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Номера листов (страниц) | | | | | Всего |  | Входящий |  |  |
| Изм | изме- | заменен- | новых | аннули- | листов | N | N сопрово- | Подп. | Да- |
| . | ненных | ных |  | рован- | (страниц) | докумен- | дительного |  | та |
|  |  |  |  | ных | в докум. | та | документа |  |  |
|  |  |  |  |  |  |  | и дата |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |