

УДОСТОВЕРЕН

ЮФКВ.10149-01-УД

**УНИВЕРСАЛЬНЫЙ ЭЛЕКТРОННЫЙ МОДУЛЬ УЭМ-МК.
ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ СИСТЕМНОЕ**

Руководство системного программиста

ЮФКВ.10149-01 32 01

(ЮФКВ.10149-01 32 01-001ФЛ)

Листов 41

Литера

Инв. N подл.	Подп. и дата	Взам. инв. N	Инв. N дубл.	Подп. и дата

АННОТАЦИЯ

Настоящий документ является руководством программиста по драйверу универсального электронного модуля мультиплексного канала УЭМ-МК (сокращённо - УЭМ) [1]. Содержит сведения, необходимые для разработки прикладных программ, описывает каждую функцию из определенных в драйвере, а также коды ошибок.

Модуль УЭМ-МК является тестером соответствия протоколу ГОСТ Р 52070-2003, обеспечивающим проведение испытаний на соответствие протоколу. Модуль УЭМ-МК также поддерживает режимы контроллера шины, оконечного устройства и монитора шины в соответствии со спецификацией ГОСТ Р 52070-2003 (см. [2]).

Данный драйвер предназначен для управления работой инструментом-мезонином УЭМ-МК. Он содержит набор функций для открытия и закрытия сеанса связи с инструментом, конфигурации и исполнения его режимов, а также ряд служебных функций. Функции драйвера разделены на классы, обеспечивающие как законченное выполнение режимов инструмента, так и управление инструментом на более низком уровне, включающее установку отдельных параметров, запуск процессов и т.п.

В документе перечислены все функции в порядке следования их в классах с описанием назначения функции, ее синтаксиса, а также с описанием каждого из параметров.

Драйвер написан на языке С в среде LabWindows/CVI 8.1 и скомпилирован как 32-разрядная динамическая библиотека DLL, предназначенная для использования в среде Windows 32-bit.

Для обеспечения доступа к устройствам на компьютере должно быть предварительно установлено программное обеспечение «VISA» и «unmbase» - драйвер носителя мезонинов.

Прикладные программы, использующие драйвер, должны разрабатываться на языке С или С++, или другом языке, обеспечивающим возможности вызова процедур на основе соглашений языка С.

Драйвер поставляется в виде комплекта следующих файлов:

- динамической библиотеки DLL;
- заголовков (.h);
- статических библиотек (.lib) связи с динамической библиотекой, предназначенные для использования с компиляторами LabWindows/CVI, Borland C++ и Visual C.

СОДЕРЖАНИЕ

1 ОБЩИЕ СВЕДЕНИЯ.....	5
2 ФУНКЦИИ ДРАЙВЕРА.....	6
2.1 Инициализация.....	7
2.2 Подключение к носителю	9
2.3 Чтение регистра.....	10
2.4 Запись в регистр	11
2.5 Чтение из FIFO.....	12
2.6 Запись в FIFO.....	13
2.7 Подталкивание FIFO	15
2.8 Подталкивание монитора шины.....	16
2.9 Запись блока в ОЗУ	16
2.10 Запрос состояния FIFO.....	18
2.11 Сброс FIFO.....	19
2.12 Сообщение МКПД	20
2.13 Команда мезонина	21
2.14 Чтение команды мезонина	22
2.15 Захват семафора.....	23
2.16 Освобождение семафора.....	23
2.17 Сброс	24
2.18 Самоконтроль	25
2.19 Запрос версии	26
2.20 Запрос ошибки	27
2.21 Запрос сообщ. об ошибке.....	28
2.22 Запрос свойств драйвера.....	30
2.23 Установка свойств драйвера.....	30
2.24 Чтение слова из ОЗУ	31
2.25 Чтение блока из ОЗУ	32
2.26 Запись слова в ОЗУ	33
2.27 Запись блока в ОЗУ	34
2.28 Номер точки трассировки	35
2.29 Получение внутреннего сеанса	36
2.30 Закрытие	37
ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ	39
ПЕРЕЧЕНЬ ССЫЛОЧНЫХ ДОКУМЕНТОВ.....	40

1 ОБЩИЕ СВЕДЕНИЯ

Для успешного использования данного драйвера должны быть соблюдены следующие условия:

- Мезонин должен быть установлен в носитель мезонинов;
- носитель мезонинов должен быть установлен в крейт VXI;
- для исполнения драйвера должен использоваться либо встроенный компьютер, либо компьютер, соединенный с крейтом VXI через интерфейс с поддержкой библиотеки VISA;
- для компиляции программы, использующей данный инструмент, должен быть использован один из компиляторов LabWindows/CVI, Borland C++ или Visual C с включением файла заголовков драйвера инструмента и соответствующей статической библиотеки в состав проекта компилятора;
- на компьютере должна быть установлена динамическая библиотека LabWindows/CVI RTE;
- на компьютере должно быть установлено программное обеспечение VISA;
- носитель мезонинов должен быть сконфигурирован менеджером ресурсов VXI.

2 ФУНКЦИИ ДРАЙВЕРА

Перечень функций драйвера с разделением по классам:

"С"-имя функции	Наименование функции/класса
unmuem_init	Инициализация
unmuem_connect	Подключение к носителю
Регистры	
unmuem_reg_read	Чтение регистра
unmuem_reg_write	Запись в регистр
FIFO	
unmuem_fifo_read	Чтение из FIFO
unmuem_fifo_write	Запись в FIFO
unmuem_fifo_push	Подталкивание FIFO
unmuem_bm_push	Подталкивание монитора шины
unmuem_block_write	Запись блока в ОЗУ
unmuem_fifo_state	Запрос состояния FIFO
unmuem_fifo_reset	Сброс FIFO
Прикладные функции (КШ)	
unmuem_bc_send	Сообщение МКПД
Управление	
unmuem_mez_cmd	Команда мезонина
unmuem_read_mez_cmd	Чтение команды мезонина
unmuem_sem_lock	Захват семафора
unmuem_sem_unlock	Освобождение семафора
Служебные Функции	
unmuem_reset	Сброс
unmuem_self_test	Самоконтроль
unmuem_revision_query	Запрос версии
unmuem_error_query	Запрос ошибки
unmuem_error_message	Запрос сообщ. об ошибке
Отладка	
unmuem_drv_caps	Запрос свойств драйвера
unmuem_set_drv_caps	Установка свойств драйвера
unmuem_dbg_read	Чтение слова из ОЗУ
unmuem_dbg_block_read	Чтение блока из ОЗУ
unmuem_dbg_write	Запись слова в ОЗУ
unmuem_dbg_block_write	Запись блока в ОЗУ
unmuem_dbg_tracepoint_q	Номер точки трассировки
unmuem_dbg_get_intssn	Получение внутреннего сеанса
unmuem_close	Закрытие

2.1 ИНИЦИАЛИЗАЦИЯ

Синтаксис

```
ViStatus _VI_FUNC unmuem_init (ViRsrc idstr, ViBoolean doID, ViBoolean doReset, ViSession *session);
```

Назначение

Данная функция производит следующие действия по инициализации инструмента:

- Открывает сеанс Менеджера Ресурсов и сеанс устройства, адрес которого указан в переменной Имя_Устройства,
- Возвращает номер сеанса связи с инструментом.

Внимание!

Для последующей работы с мезонином необходимо полученный при инициализации сеанс привязать к открытому сеансу носителя мезонинов, полученному из функции unmbase_init(). Это делается с помощью функции подключения к носителю (_connect())

Без подключения к носителю мезонинов никакая дальнейшая работа не возможна.

Пример начальной последовательности действий.

```
unmbase_init("VXI0::1::INSTR", VI_TRUE, VI_TRUE, &vi);  
unmuem_init("VXI0::1::INSTR", VI_TRUE, VI_TRUE, &mvi);  
unmuem_connect(mvi, vi, num, VI_TRUE, VI_TRUE);
```

Список параметров

<idstr>

тип переменной ViRsrc

Данная переменная указывает тип интерфейса и адрес устройства для инструмента, с которым открывается сеанс. Грамматические правила определения данной строки приведены в примечаниях.

Значение по умолчанию: "VXI0::1::INSTR"

Примечания:

Грамматические правила описания адреса инструмента. Необязательные параметры приведены в квадратных скобках.

Интерфейс Грамматика строки адреса

VXI [board]::VXI logical address[::INSTR]

По умолчанию значение параметра board=0.

<doID>

тип переменной ViBoolean

Данный параметр определяет, производить ли идентификацию инструмента в процедуре инициализации.

Допустимые значения:

VI_OFF (0) - Не производить идентификацию

VI_ON (1) - Производить идентификацию (по умолчанию)

Примечание:

Обычно не следует отключать идентификацию инструмента, так как это дает дополнительную проверку на соответствие типа инструмента тому, на который рассчитан драйвер.

<doReset>

тип переменной ViBoolean

Данный параметр определяет, производить ли сброс инструмента в процедуре инициализации.

Допустимые значения:

VI_OFF (0) - Не производить сброс

VI_ON (1) - Производить сброс (по умолчанию)

<session>

тип переменной ViSession (указатель)

В данной переменной функция возвращает уникальный идентификатор сеанса связи с инструментом, который необходимо сохранить для всех последующих вызовов функций драйвера инструмента.

Примечание:

При каждом новом вызове функции инициализации открывается еще один сеанс связи с тем же инструментом.

Возвращаемое значение:

тип переменной ViStatus

Возвращаемое функцией значение может содержать как код ошибки, так и код завершения, причем коды ошибок принимают значения меньше нуля.

Возможные ошибки:

UNMUEM_ERROR_NO_HOST_MEMORY

Ошибки VISA.

Расшифровка кодов ошибок приведена в описании функции

unmuem_error_message()

2.2 ПОДКЛЮЧЕНИЕ К НОСИТЕЛЮ

Синтаксис

```
ViStatus _VI_FUNC unmuem_connect (ViSession mvi, ViSession vi, ViUInt16 meznum, ViBoolean idn, ViBoolean reset);
```

Назначение

Данная функция готовит к работе сеанс мезонина, открытый ранее с помощью функции инициализации. Данную функцию необходимо вызывать один раз сразу после инициализации мезонина.

Функция завершает конфигурирование сеанса мезонина. В том числе - конфигурируются FIFO для передачи данных из мезонина и в мезонин. (Конфигурирование FIFO можно отключить при помощи функции `unmuem_set_drv_caps()` (см..))

Список параметров

<mvi>

тип переменной ViSession

В данном параметре указывается номер сеанса связи мезонина, возвращенный функцией инициализации.

<vi>

тип переменной ViSession

В данном параметре указывается номер сеанса носителя, на котором установлен мезонин. Сеанс носителя возвращается функцией инициализации драйвера носителя мезонинов (`unmbase_init()`).

<meznum>

тип переменной ViUInt16

В данном параметре указывается позиция мезонина на носителе мезонинов. Определить типы установленных на носителе мезонинов можно с помощью функции `unmbase_m_type_q()` драйвера носителя.

<idn>

тип переменной ViBoolean

Данный параметр определяет, производить ли идентификацию инструмента.

Допустимые значения:

VI_OFF (0) - Не производить идентификацию

VI_ON (1) - Производить идентификацию (по умолчанию)

Примечание:

Обычно не следует отключать идентификацию инструмента, так как это дает дополнительную проверку на соответствие типа инструмента тому, на который рассчитан драйвер.

<reset>

тип переменной ViBoolean

Данный параметр определяет, производить ли сброс инструмента в процедуре инициализации.

Допустимые значения:

VI_OFF (0) - Не производить сброс

VI_ON (1) - Производить сброс (по умолчанию)

Возвращаемое значение:

тип переменной ViStatus

Возвращаемое функцией значение может содержать как код ошибки, так и код завершения, причем коды ошибок принимают значения меньше нуля.

VI_ERROR_INV_OBJECT

UNMUEM_ERROR_TIMER_CONF

UNMUEM_ERROR_THREAD_START

Ошибки VISA.и UNMBASE

Расшифровка кодов ошибок приведена в описании функции

unmuem_error_message()

2.3 ЧТЕНИЕ РЕГИСТРА

Синтаксис

```
ViStatus _VI_FUNC unmuem_reg_read (ViSession mvi, ViUInt16 reg, ViUInt32 *value);
```

Назначение

Чтение регистра мезонина

Список параметров

<mvi>

тип переменной ViSession

Сеанс мезонина

<reg>

тип переменной ViUInt16

Номер регистра

<value>

тип переменной ViUInt32 (указатель)

Данные из регистра

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

UNMUEM_ERROR_INVALID_ADDR

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции
unmuem_error_message()

2.4 ЗАПИСЬ В РЕГИСТР

Синтаксис

```
ViStatus _VI_FUNC unmuem_reg_write (ViSession mvi, ViUInt16 reg, ViUInt32 value);
```

Назначение

Запись в регистр мезонина

Список параметров

<mvi>

тип переменной ViSession

Сеанс мезонина

<reg>

тип переменной ViUInt16

Номер регистра

<value>

тип переменной ViUInt32

Данные для записи в регистр

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки^

VI_ERROR_INV_OBJECT

UNMUEM_ERROR_INVALID_ADDR

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции

unmuem_error_message()

2.5 ЧТЕНИЕ ИЗ FIFO

Синтаксис

```
ViStatus _VI_FUNC unmuem_fifo_read (ViSession mvi, ViUInt32 message[], ViUInt32 nframes,  
ViUInt32 *fnframes);
```

Назначение

Считывание данных из FIFO от мезонина к управляющему вычислителю.

Список параметров

<mvi>

тип переменной ViSession

Сеанс мезонина

<message>

тип переменной ViUInt32 (указатель)

Массив-буфер для приема данных из FIFO.

Данные в FIFO представляют собой последовательность "кадров",
каждый "кадр" состоит из 2х 32-битных слов. В первом слове:

бит 31: 0 означает последний кадр в сообщении, 1 - не последний,
бит 30: 1 означает незначущий кадр (заполнитель),
биты 29-23 не используются,
бит 22: переключатель регистры(0) / ОЗУ (1)
биты 21-0: адрес в ОЗУ или номер регистра.
Второе слово - слово данных.

<nframes>

тип переменной ViUInt32

Число кадров (2x32 бит), которое необходимо прочитать.
Считываемые данные должны помещаться в массив message.

<fnframes>

тип переменной ViUInt32 (указатель)

Число фактически прочитанных кадров.

Если в FIFO меньше непрочитанных кадров, чем запрошено,
будет считаны имеющиеся кадры.

Если в FIFO нет непрочитанных данных,
будет возвращено значение 0.

Даже в случае обнаружения ошибки функция возвращает корректное значение числа фактически прочитанных кадров, и оно может быть ненулевым.

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции
unmuem_error_message()

2.6 ЗАПИСЬ В FIFO

Синтаксис

```
ViStatus _VI_FUNC unmuem_fifo_write (ViSession mvi, ViUInt32 message[], ViUInt32 nframes,  
ViUInt32 *fnframes);
```

Назначение

Запись данных в FIFO от управляющего вычислителя к мезонину.

ВНИМАНИЕ: фактическая передача данных в мезонин начинается после вызова функции `unmuem_fifo_push()` (см.)

Список параметров

<mvi>

тип переменной `ViSession`

Сеанс мезонина

<message>

тип переменной `ViUInt32` (указатель)

Массив, содержащий сообщение, подготовленное в соответствии с "протоколом FIFO".

Сообщение состоит из целого числа "кадров",

каждый "кадр" - из 2х 32-битных слов. В первом слове:

бит 31: 0 означает последний кадр в сообщении, 1 - не последний,

бит 30: 1 означает незначащий кадр (заполнитель),

биты 29-23 не используются,

бит 22: переключатель регистры(0) / ОЗУ (1)

биты 21-0: адрес в ОЗУ или номер регистра.

Второе слово - слово данных.

<nframes>

тип переменной `ViUInt32`

Размер сообщения, в кадрах (2х32 бита).

<fnframes>

тип переменной `ViUInt32` (указатель)

Число фактически записанных кадров. Это число может быть меньше, чем <nframes>, если в FIFO недостаточно места для размещения данных. В текущей версии драйвера размер FIFO составляет 60Мбайт, или 7,5М кадров.

Даже в случае обнаружения ошибки функция возвращает корректное значение числа фактически записанных кадров, и оно может быть ненулевым.

Возвращаемое значение:

тип переменной `ViStatus`

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

`VI_ERROR_INV_OBJECT`

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции
unmuem_error_message()

2.7 ПОДТАЛКИВАНИЕ FIFO

Синтаксис

```
ViStatus _VI_FUNC unmuem_fifo_push (ViSession mvi);
```

Назначение

Данная функция запускает передачу данных в мезонин. Данные должны быть до этого записаны в FIFO при помощи функции unmuem_fifo_write() (см.)

ВНИМАНИЕ: приложение НЕ ОБЯЗАНО дожидаться завершения передачи данных в мезонин, и может немедленно начать добавление в FIFO следующей порции данных.

Выяснить, закончена ли передача данных в мезонин можно при помощи функции unmuem_fifo_state() (см.).

ДОПОЛНИТЕЛЬНО: в драйвере реализована функция "autopush", которая запускает передачу данных в мезонин автоматически при накоплении в FIFO некоторого объема данных. Пороговый объем данных для автоматического запуска передачи данных может отличаться в разных версиях драйвера. В текущей версии он равен 333 кадрам. Отключить функцию "autopush" можно при помощи функции unmuem_set_drv_caps() (см.).

Список параметров

<mvi>

тип переменной ViSession

Сеанс мезонина

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции
unmuem_error_message()

2.8 ПОДТАЛКИВАНИЕ МОНИТОРА ШИНЫ

Синтаксис

```
ViStatus _VI_FUNC unmuem_bm_push (ViSession mvi);
```

Назначение

Данная функция вызывает запись данных монитора шины в fifo (для передачи от УЭМ к ПЭВМ), не дожидаясь заполнения внутреннего буфера монитора шины. Имеет смысл только при сконфигурированном и запущенном режиме монитора шины.

Список параметров

<mvi>

тип переменной ViSession

Сеанс мезонина

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции

unmuem_error_message()

2.9 ЗАПИСЬ БЛОКА В ОЗУ

Синтаксис

```
ViStatus _VI_FUNC unmuem_block_write (ViSession mvi, ViUInt32 address, ViUInt32 data[],  
ViUInt32 nwords, ViUInt32 *fnwords);
```

Назначение

Эта функция выполняет запись блока данных в ОЗУ мезонина по указанному адресу. Она формирует соответствующее сообщение протокола FIFO и передает его в функцию `unmuem_fifo_write`.

Список параметров

<mvi>

тип переменной ViSession

Сеанс мезонина

<address>

тип переменной ViUInt32

Начальный адрес блока в ОЗУ мезонина

<data>

тип переменной ViUInt32 (указатель)

Через эту переменную передается адрес блока данных, записываемых в ОЗУ, Блок должен состоять из целого числа 32-битных слов

<nwords>

тип переменной ViUInt32

Размер блока данных в 32-битных словах

<fnwords>

тип переменной ViUInt32 (указатель)

Число фактически записанных слов данных.

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

UNMUEM_ERROR_NO_HOST_MEMORY

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции

`unmuem_error_message()`

2.10 ЗАПРОС СОСТОЯНИЯ FIFO

Синтаксис

```
ViStatus _VI_FUNC unmuem_fifo_state (ViSession mvi, ViUInt16 direction, ViInt16 *State);
```

Назначение

Функция предназначена для запроса состояния FIFO ввода и вывода: выполняется ли обмен данными или FIFO остановлена.

Список параметров

<mvi>

тип переменной ViSession

Сеанс мезонина

<direction>

тип переменной ViUInt16

В данном параметре указывается конкретная FIFO:

0 - FIFO на чтение данных из мезонина

1 - FIFO на запись данных в мезонин

<State>

тип переменной ViInt16 *

В этом параметре возвращается состояние указанной FIFO: 0 - FIFO остановлена, 1 - FIFO выполняет передачу данных из (в) мезонин. В нормальном состоянии для FIFO ввода из мезонина всегда возвращается 1. Для FIFO вывода в мезонин значение 1 указывает, что в FIFO имеются данные для вывода и они передаются в мезонин.

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции

unmuem_error_message()

2.11 СБРОС FIFO

Синтаксис

ViStatus _VI_FUNC unmuem_fifo_reset (ViSession mvi, ViUInt16 direction);

Назначение

Эта функция выполняет остановку обмена и сброс FIFO от мезонина к УВ или от УВ к мезонина (в зависимости от параметра direction).

В нормальном состоянии устройства обмен через FIFO от мезонина к УВ постоянно запущен, поэтому после сброса он автоматически запускается снова. См. подробнее unmuem_connect(), unmuem_set_drv_caps().

.

Список параметров

<mvi>

тип переменной ViSession

Сеанс мезонина

<direction>

тип переменной ViUInt16

В данном параметре указывается конкретная FIFO:

0 - FIFO на чтение данных из мезонина

1 - FIFO на запись данных в мезонин

2 - обе

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции unmuem_error_message()

2.12 СООБЩЕНИЕ МКПД

Синтаксис

```
ViStatus _VI_FUNC unmuem_bc_send (ViSession mvi, ViUInt16 rt, ViBoolean rctx, ViUInt16 sa,  
ViUInt16 ndw_mc, ViUInt16 _VI_FAR dw[], ViBoolean channel, ViUInt16 *statusword);
```

Назначение

Данная функция посылает сообщение МКПД, используя аппаратуру КШ в составе УЭМ, и принимает ответ, используя аппаратуру МШ в составе УЭМ. Позволяет программно сформировать командный сегмент. Поддерживает форматы сообщений 1,2, 4-7, 9, 10 ГОСТ Р 52070-2003

Список параметров

<mvi>

тип переменной ViSession

Сеанс мезонина

<rt>

тип переменной ViUInt16

Номер ОУ

<rctx>

тип переменной ViBoolean

Команда: Передать (TX (1)) или принять (RX (0)) данные

<sa>

тип переменной ViUInt16

Подадрес

<ndw_mc>

тип переменной ViUInt16

Число слов данных для обычных команд и код управления для команд управления

<dw>

тип переменной ViUInt16 (указатель)

Этот массив используется как входной или выходной буфер для слов данных сообщения МКПД - в зависимости от формата сообщения МКПД. Должен быть не менее 32 16-битных слов.

<channel>

тип переменной ViBoolean

Этот параметр определяет шину, по которой будет отправлено сообщение: А (0) или Б (1)

<statusword>

тип переменной ViUInt16 (указатель)

Через этот параметр возвращается ответное слово МКПД

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

UNMUEM_ERROR_INCORRECT_MESSAGE_FORMAT

UNMUEM_ERROR_WRITE

UNMUEM_ERROR_NO_HOST_MEMORY

UNMUEM_ERROR_NO_RESPONSE

UNMUEM_ERROR_INVALID_RESPONSE

UNMUEM_WARN_MESSAGE_ERROR

UNMUEM_WARN_STATUS_SET

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции

unmuem_error_message()

2.13 КОМАНДА МЕЗОНИНА

Синтаксис

ViStatus _VI_FUNC unmuem_mez_cmd (ViSession mvi, ViUInt16 cmd);

Назначение

Запись кода в регистр команд мезонина MEZ_CMD

Список параметров

<mvi>

тип переменной ViSession

В данном параметре передается сеанс связи с инструментом, возвращенный функцией инициализации.

<cmd>

тип переменной ViUInt16

Код команды мезонина. Коды команд приведены в документации по УЭМ

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции unmuem_error_message()

2.14 ЧТЕНИЕ КОМАНДЫ МЕЗОНИНА

Синтаксис

```
ViStatus _VI_FUNC unmuem_read_mez_cmd (ViSession mvi, ViUInt16 *cmd);
```

Назначение

Чтение кода из регистра команд мезонина MEZ_CMD

Список параметров

<mvi>

тип переменной ViSession

В данном параметре передается сеанс связи с инструментом, возвращенный функцией инициализации.

<cmd>

тип переменной ViUInt16 *

Через это параметр передается считанное значение регистра

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции
unmuem_error_message()

2.15 ЗАХВАТ СЕМАФОРА

Синтаксис

```
ViStatus _VI_FUNC unmuem_sem_lock (ViSession mvi);
```

Назначение

Захват аппаратного семафора (mez_sem в регистре mez_cmd)

Список параметров

<mvi>

тип переменной ViSession

В данном параметре передается сеанс связи с инструментом, возвращенный функцией инициализации.

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

UNMUEM_WARN_SEM_REJECT - отказ захвата семафора, это признак того, что семафор уже захвачен (этой же или другой задачей)

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции
unmuem_error_message()

2.16 ОСВОБОЖДЕНИЕ СЕМАФОРА

Синтаксис

ViStatus _VI_FUNC unmuem_sem_unlock (ViSession mvi);

Назначение

Освобождение аппаратного семафора (mez_sem в регистре mez_cmd)

Список параметров

<mvi>

тип переменной ViSession

В данном параметре передается сеанс связи с инструментом, возвращенный функцией инициализации.

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции

unmuem_error_message()

2.17 СБРОС

Синтаксис

ViStatus _VI_FUNC unmuem_reset (ViSession mvi);

Назначение

Данная функция производит сброс инструмента в состояние по умолчанию.

Список параметров

<mvi>

тип переменной ViSession

В данном параметре передается сеанс связи с инструментом, возвращенный функцией инициализации.

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции

unmuem_error_message()

2.18 САМОКОНТРОЛЬ

Синтаксис

```
ViStatus _VI_FUNC unmuem_self_test (ViSession mvi, ViInt16 *res, ViChar _VI_FAR msg[]);
```

Назначение

Данная функция производит самоконтроль инструмента и возвращает его результат.

Список параметров

<mvi>

тип переменной ViSession

В данном параметре передается сеанс связи с инструментом, возвращенный функцией инициализации.

<res>

тип переменной ViInt16 (указатель)

Данная переменная содержит значение, соответствующее результату самоконтроля.

Нулевое значение означает успешное прохождение самоконтроля.

Прочие коды свидетельствуют об ошибке.

Возможные результаты и соответствующие сообщения :

0 - Тест самоконтроля успешно выполнен

-1 - Отказ при выполнении теста самоконтроля

-2 - Тест самоконтроля выявил ошибки

<msg>

тип переменной ViChar (указатель)

Данная переменная содержит строку сообщения о результате самоконтроля. Возвращаемые строки приведены в описании параметра res.

Примечание:

Строка должна содержать 256 элементов ViChar[256].

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

UNMUEM_ERROR_NO_HOST_MEMORY

UNMUEM_ERROR_WRITE

UNMUEM_WARN_SELF_TEST_FAILED

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции

unmuem_error_message()

2.19 ЗАПРОС ВЕРСИИ

Синтаксис

```
ViStatus _VI_FUNC unmuem_revision_query (ViSession mvi, ViChar _VI_FAR dv[], ViChar _VI_FAR iv[]);
```

Назначение

Данная функция возвращает версию драйвера и микропрограммы инструмента.

Список параметров

<mvi>

тип переменной ViSession

В данном параметре передается сеанс связи с инструментом, возвращенный функцией инициализации.

Примечание:

В данной функции в этом параметре допускается передавать VI_NULL.

<dv>

тип переменной ViChar (указатель)

В данной строке возвращается версия драйвера инструмента.

Примечание:

Строка должна содержать 256 элементов ViChar[256].

<iv>

тип переменной ViChar (указатель)

В данной строке возвращается версия инструмента.

Примечания:

(1) Строка должна содержать 256 элементов ViChar[256].

(2) Если в функцию передан сеанс NULL, будет возвращена строка "n/a"

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

Расшифровка кодов ошибок приведена в описании функции
unmuem_error_message()

2.20 ЗАПРОС ОШИБКИ

Синтаксис

```
ViStatus _VI_FUNC unmuem_error_query (ViSession mvi, ViInt32 *status, ViChar _VI_FAR msg[]);
```

Назначение

Возвращает код ошибки последней операции и соответствующее текстовое сообщение. Функция не реализована.

Список параметров

<mvi>

тип переменной ViSession

В данном параметре передается сеанс связи с инструментом, возвращенный функцией инициализации.

<status>

тип переменной ViInt32 (указатель)

В данной переменной возвращается код ошибки, считанный из очереди ошибок инструмента.

<msg>

тип переменной ViChar (указатель)

В данной строке возвращается сообщение об ошибке, соответствующее коду ошибки.

Примечание:

Строка должна содержать 256 элементов ViChar[256].

Возвращаемое значение:

тип переменной ViStatus

Код ошибки. Всегда VI_ERROR_NIMPL_OPER

2.21 ЗАПРОС СООБЩ. ОБ ОШИБКЕ

Синтаксис

```
ViStatus _VI_FUNC unmuem_error_message (ViSession mvi, ViStatus status, ViChar _VI_FAR msg[]);
```

Назначение

Данная функция получает и интерпретирует код состояния, возвращенный какой-либо функцией драйвера, и возвращает строку сообщения об ошибке. Если код состояния сгенерирован не данным драйвером, функция обращается для получения строки сообщения к аналогичной функции драйвера unmbase

Возможные значения кодов ошибок:

Код	Определение
VI_ERROR_NIMPL_OPER	Нереализованная операция
VI_ERROR_INV_OBJECT	Недействительный объект (чаще всего - недействительный сеанс)
UNMUEM_ERROR_INVALID_ADDR	Недействительный адрес
UNMUEM_ERROR_NO_HOST_MEMORY	Недостаточно ОЗУ ПЭВМ
UNMUEM_ERROR_TIMER_CONF	Отказ конфигурирования таймера
UNMUEM_ERROR_THREAD_START	Отказ запуска нити
UNMUEM_ERROR_READ	Ошибка чтения данных
UNMUEM_ERROR_WRITE	Ошибка записи данных
UNMUEM_ERROR_INCORRECT_MESSAGE_FORMAT	Некорректный формат

сообщения

UNMUEM_ERROR_NO_RESPONSE	Нет ответа ОУ
UNMUEM_ERROR_INVALID_RESPONSE	Недостоверный ответ ОУ
UNMUEM_WARN_SEM_REJECT	Семафор уже закрыт
UNMUEM_WARN_PRIORITY_CONF	Ошибка конфигурирования приоритета
UNMUEM_WARN_SELF_TEST_FAILED	Тест самоконтроля выявил ошибки
UNMUEM_WARN_MESSAGE_ERROR	Установлен статус ОИШС
UNMUEM_WARN_STATUS_SET	Установлены биты статуса (кроме ОИШС)

За расшифровкой кодов ошибок, сгенерированных подсистемами VISA и unmbase следует обращаться к документации по этим подсистемам

Список параметров

<mvi>

тип переменной ViSession

В данном параметре передается сеанс связи с инструментом, возвращенный функцией инициализации.

Примечание:

В данной функции в этом параметре допускается передавать VI_NULL.

<status>

тип переменной ViStatus

Данная переменная должна содержать код состояния, возвращенный какой-либо из функций драйвера.

<msg>

тип переменной ViChar (указатель)

В данной строке возвращается сообщение, соответствующее переданному коду состояния.

Примечание:

Строка должна содержать 256 элементов ViChar[256].

Возвращаемое значение:

тип переменной ViStatus

Возвращаемое функцией значение содержит либо VI_SUCCESS, если код ошибки интерпретирован успешно, либо предупреждение VI_WARN_UNKNOWN_STATUS, если код ошибки неизвестен.

2.22 ЗАПРОС СВОЙСТВ ДРАЙВЕРА

Синтаксис

```
ViUInt32 _VI_FUNC unmuem_drv_caps()
```

Назначение

Возвращает свойства драйвера (строку бит). Свойства драйвера - это настраиваемые и ненастраиваемые параметры драйвера, предназначенные для отладки драйвера и аппаратуры. Эти параметры относятся к драйверу в целом и влияют на все открытые устройства.

Возвращаемое значение:

тип переменной ViUInt32

Возвращаются свойства драйвера - битовая строка, объединение по OR следующих констант:

UNMUEM_SIMDRIVER (=1) - данный драйвер является драйвером программного симулятора, а не реального устройства..

UNMUEM_AUTOFIFO (=2) - при установленном бите в функциим unmuem_connect() выполняется конфигурирование FIFO, при сброшенном - не выполняется

UNMUEM_AUTOPUSH (=4) - включена функция "autopush"

2.23 УСТАНОВКА СВОЙСТВ ДРАЙВЕРА

Синтаксис

```
void _VI_FUNC unmuem_set_drv_caps(ViUInt32 caps);
```

Назначение

Устанавливает свойства драйвера (строку бит). Свойства драйвера - это настраиваемые и ненастраиваемые параметры драйвера, предназначенные для отладки драйвера и аппаратуры. Эти параметры относятся к драйверу в целом и влияют на все открытые устройства.

Список параметров

<caps>

тип переменной ViUInt32

В этом параметре передются свойства драйвера - битовая строка, объединение по OR следующих констант:

UNMUEM_SIMDRIVER (=1) - данный драйвер является драйвером программного симулятора, а не реального устройства. Значение этого свойства определяется редакцией драйвера и не может быть изменено.

UNMUEM_AUTOFIFO (=2) - при установленном бите в фунцкмм unmuem_connect() выполняется конфигурирование FIFO, при сброшенном - не выполняется

UNMUEM_AUTOPUSH (=4) - включена функция "autopush"

При нормальном конфигурировании драйвера свойства UNMUEM_AUTOFIFO и UNMUEM_AUTOPUSH установлены (включены). Режим работы драйвера со сброшенными (выключенными) свойствами UNMUEM_AUTOFIFO и UNMUEM_AUTOPUSH предназначен для отладки аппаратуры и самого драйвера.

2.24 ЧТЕНИЕ СЛОВА ИЗ ОЗУ

Синтаксис

```
ViStatus _VI_FUNC unmuem_dbg_read (ViSession mvi, ViUInt32 address, ViUInt32 *data);
```

Назначение

Эта функция выполняет считывание 1 32-битного слова из ОЗУ устройства. Использует аппаратный интерфейс отладочного считывания. Интерфейс является медленным и не предназначен для использования под рабочей нагрузкой.

Список параметров

<mvi>

тип переменной ViSession

Сеанс мезонина

<address>

тип переменной ViUInt32

Адрес слова в ОЗУ мезонина

<data>

тип переменной ViUInt32 *

Через эту переменную передается адрес слова, в которое будет считываться значение слова из ОЗУ устройства

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции
unmuem_error_message()

2.25 ЧТЕНИЕ БЛОКА ИЗ ОЗУ

Синтаксис

```
ViStatus _VI_FUNC unmuem_dbg_block_read (ViSession mvi, ViUInt32 address, ViUInt32 _VI_FAR  
data[], ViUInt32 nwords);
```

Назначение

Эта функция выполняет считывание блока данных из ОЗУ устройства. Использует аппаратный интерфейс отладочного считывания. Интерфейс является медленным и не предназначен для использования под рабочей нагрузкой.

Список параметров

<mvi>

тип переменной ViSession

Сеанс мезонина

<address>

тип переменной ViUInt32

Начальный адрес блока в ОЗУ мезонина

<data>

тип переменной ViUInt32 (указатель)

Через эту переменную передается адрес буфера данных для считывания из ОЗУ, Блок должен состоять из целого числа 32-битных слов

<nwords>

тип переменной ViUInt32

Размер блока данных в 32-битных словах

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции
unmuem_error_message()

2.26 ЗАПИСЬ СЛОВА В ОЗУ

Синтаксис

ViStatus _VI_FUNC unmuem_dbg_write (ViSession mvi, ViUInt32 address, ViUInt32 data);

Назначение

Эта функция выполняет запись 1 32-битного слова из ОЗУ устройства. Использует аппаратный интерфейс отладочной записи. Интерфейс является медленным и не предназначен для использования под рабочей нагрузкой.

Список параметров

<mvi>

тип переменной ViSession

Сеанс мезонина

<address>

тип переменной ViUInt32

Адрес слова в ОЗУ мезонина

<data>

тип переменной ViUInt32

Значение, которое надо записать в слово

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции
unmuem_error_message()

2.27 ЗАПИСЬ БЛОКА В ОЗУ

Синтаксис

```
ViStatus _VI_FUNC unmuem_dbg_block_write (ViSession mvi, ViUInt32 address, ViUInt32 _VI_FAR  
data[], ViUInt32 nwords);
```

Назначение

Эта функция выполняет запись блока данных из ОЗУ устройства. Использует аппаратный интерфейс отладочной записи. Интерфейс является медленным и не предназначен для использования под рабочей нагрузкой.

Список параметров

<mvi>

тип переменной ViSession

Сеанс мезонина

<address>

тип переменной ViUInt32

Начальный адрес блока в ОЗУ мезонина

<data>

тип переменной ViUInt32 (указатель)

Через эту переменную передается адрес буфера с данными для записи в ОЗУ, Блок должен состоять из целого числа 32-битных слов

<nwords>

тип переменной ViUInt32

Размер блока данных в 32-битных словах

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции
unmuem_error_message()

2.28 НОМЕР ТОЧКИ ТРАССИРОВКИ

Синтаксис

```
int _VI_FUNC unmuem_dbg_tracepoint_q)(
```

Назначение

Возвращает номер точки трассировки.

Номер точки трассировки выставляется функциями init и connect.

Номер точки трассировки предназначен для определения, на какой стадии исполнения функции произошла ошибка. Номера:

INIT_ALLOC = 0,

INIT_OPEN_DEFRM = 1,

INIT_OPEN = 2,

INIT_SET_ATTR = 3,

CONNECT_GET_ATTR = 4,

```
CONNECT_M_INIT = 5,  
CONNECT_SET_ATTR_1 = 6,  
CONNECT_SET_ATTR_2 = 7,  
CONNECT_TYPE_Q = 8,  
CONNECT_CHECK_TYPE = 9,  
CLOSE_GET_ATTR = 10;
```

Этот механизм использует глобальную переменную и принципиально не реентерабельный.

Возвращаемое значение:

тип переменной int

2.29 ПОЛУЧЕНИЕ ВНУТРЕННЕГО СЕАНСА

Синтаксис

```
ViStatus _VI_FUNC unmuem_dbg_get_intssn (ViSession mvi, ViSession *intvi);
```

Назначение

Функция возвращает "внутренний" сеанс мезонина. Этот сеанс может быть использован для автономной разработки и отладки дополнительных функций драйвера.

Список параметров

<mvi>

тип переменной ViSession

Сеанс мезонина

<intvi>

тип переменной ViSession (указатель)

Внутренний сеанс мезонина

Возвращаемое значение:

тип переменной ViStatus

Код завершения(0, >=0) или ошибки (<0).

Возможные ошибки:

VI_ERROR_INV_OBJECT

Ошибки VISA и UNMBASE

Расшифровка кодов ошибок приведена в описании функции

unmuem_error_message()

2.30 ЗАКРЫТИЕ

Синтаксис

```
ViStatus _VI_FUNC unmuem_close (ViSession mvi);
```

Назначение

Данная функция закрывает сеанс связи с инструментом и сеанс связи с менеджером ресурсов.

Примечание:

после вызова данной функции для использования инструмента вновь надо вызвать функцию инициализации.

Список параметров

<mvi>

тип переменной ViSession

В данном параметре передается сеанс связи с инструментом, возвращенный функцией инициализации.

Возвращаемое значение:

тип переменной ViStatus

Возвращаемое функцией значение может содержать как код ошибки, так и код завершения, причем коды ошибок принимают значения меньше нуля. Возможные значения кодов ошибок приведены в описании функции запроса сообщения об ошибке.

ПЕРЕЧЕНЬ ПРИНЯТЫХ СОКРАЩЕНИЙ И ОБОЗНАЧЕНИЙ

BC	- режим контроллера в терминах интерфейса [2] – синоним КШ (Bus Controller)
RT	- режим оконечного устройства в терминах интерфейса [2] – синоним ОУ (Remote Terminal)
MM	- режим избирательного монитора сообщений (Selective Message Monitor)
MT	режим Монитора в Терминах интерфейса [2]
КШ	- режим контроллера в терминах интерфейса [2]
МКПД	- мультиплексный канал передачи данных
УЭМ	- универсальный электронный модуль
ПО	- программное обеспечение
П/П	- подпрограмма
ПЭВМ	- персональная электронная вычислительная машина
РАОУ	- регистр адреса оконечного устройства
ТСТ	- режим тестера
УВ	- управляющий вычислитель

ПЕРЕЧЕНЬ ССЫЛОЧНЫХ ДОКУМЕНТОВ

1. ЮФКВ.469555.555РЭ Универсальный электронный модуль УЭМ-МК. Руководство по эксплуатации
2. ГОСТ Р 52070-2003 Интерфейс магистральной последовательной системы электронных модулей. Общие требования
(Примечание: до 01.01.2004 – ГОСТ 26765.52-87)

