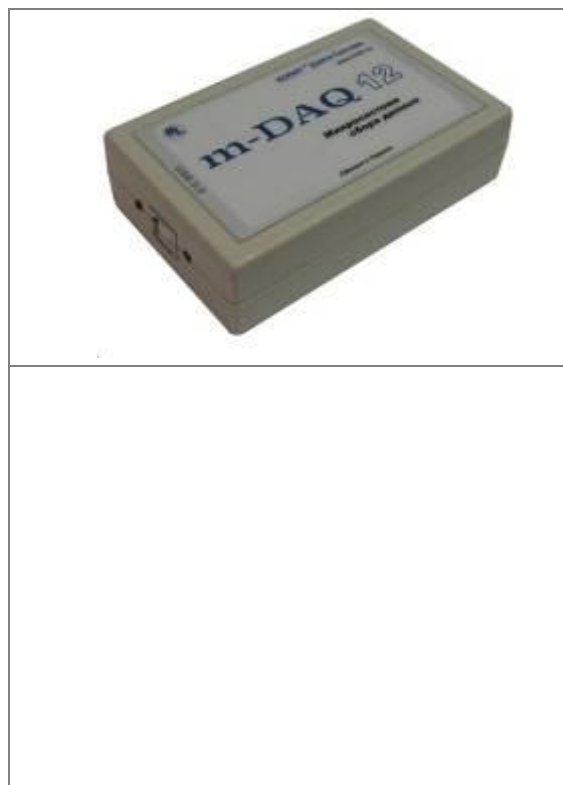

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ

V1.4

m-DAQ12 m-DAQ14

микросистема сбора данных с интерфейсом USB



ВВЕДЕНИЕ

Микросистемы сбора данных m-DAQ12 и m-DAQ14 предназначены для ввода/вывода аналоговых и дискретных сигналов. Микросистемы могут быть применены как в составе мобильных измерительных комплексов, так и в лабораторных или промышленных условиях.

m-DAQ12 и m-DAQ14 являются многофункциональными измерительными модулями, подсоединяемыми к ПК через USB-интерфейс (USB 2.0 HighSpeed). Подключение сигналов осуществляется через разъем, расположенный на боковой стенке модуля.

Микросистемы включают в себя:

- 12-ти разрядный (в m-DAQ12) либо 14-ти разрядный (в m-DAQ14) аналогово-цифровой преобразователь с мультиплексированием входных аналоговых каналов;

- порт ввода-вывода дискретных сигналов;
- 16-разрядный счетчик;
- 2-х канальный 12-ти разрядный цифро-аналоговый преобразователь (опция).

Микросистемы позволяют работать с 4-мя дифференциальными или 8-ю каналами с общей землей для ввода аналоговых сигналов.

Каждый из аналоговых каналов подключается к АЦП через программно управляемый усилитель, позволяющий задавать один из четырех диапазонов измерения напряжения (± 10 В, ± 5 В, ± 2.5 В, ± 1.25 В).

Модули обеспечивают непрерывный (синхронизированный по внутреннему таймеру) сбор данных на частотах дискретизации АЦП от 65 Гц до 100 КГц.

Цифровой ввод-вывод представлен в виде 8-ми цифровых линий, индивидуально конфигурируемых на ввод или вывод и совместимых с ТТЛ уровнями.

На контактах внешнего разъема присутствуют напряжения питания для внешних устройств +5 В и ± 15 В.

Базовые модели микросистем m-DAQ12 и m-DAQ14 выполнены в виде модуля в компактном корпусе из АВС-пластика, подключение к внешним устройствам осуществляется через разъем DB-25. OEM-версии микросистем, выполненные в виде платы, ориентированы на встраиваемые применения. Для подключения к внешним устройствам в ней используется двухрядный штыревой соединитель типа IDC-26.

Программное обеспечение микросистем содержит драйверы, DLL-библиотеку и примеры работы с ней, Iib-библиотеку для среды графического программирования LabVIEW, и программу «Осциллограф-регистратор».

Комплект поставки m-DAQ12:

1. Модуль m-DAQ12;
2. Кабель связи по USB тип A-B, 28AWG/1PR-24AWG/2C, 1.8...2.0 м;
3. Кабельная часть разъема DB-25M;
4. CD-диск с программным обеспечением и руководством пользователя.

Комплект поставки m-DAQ14:

1. Модуль m-DAQ14;
2. Кабель связи по USB тип A-B, 28AWG/1PR-24AWG/2C, 1.8...2.0 м;
3. Кабельная часть разъема DB-25M;
4. CD-диск с программным обеспечением и руководством пользователя.

ИНСТАЛЯЦИЯ**Системные требования**

| | Для корректной работы устройства | Для корректной работы под NI LabVIEW 7.1 и выше |
|----------------------------------|--|---|
| Процессор | не ниже Pentium III | не ниже Pentium IV |
| Интерфейс | USB 2.0 с возможностью работы в режиме HighSpeed (480Mb/s) | |
| Оперативная память | не менее 512 MB | |
| Свободное место на жестком диске | не менее 10 MB | не менее 100 MB |
| Операционная система | не ниже Microsoft® Windows XP | |

Порядок подключения к ПК

Спецификация USB разрешает как горячее подключение устройств к шине USB, с их автоматическим распознаванием, так и включение ПК с уже подключенным внешним модулем.

- включить ПК и загрузить одну из операционных систем, поддерживающих работу с USB-устройством;
- соединить USB разъем модуля и свободный USB-порт ПК с помощью USB-кабеля, входящего в комплект поставки.

Установка драйвера

Шина USB предоставляет пользователям возможность работать с периферийными устройствами в режиме Plug & Play. Инициализация программных драйверов шины осуществляется операционной системой после распознавания нового устройства.

При первом подсоединении модуля к ПК операционная система запросит файлы драйвера для подключенного устройства. Пользователю необходимо указать расположение файла **mdaq2ldr.inf** и **mdaq2.inf**. В случае успешной инициализации информация о драйвере будет занесена в реестр Windows, и при повторных сеансах работы устройство будет инициализироваться автоматически. В случае необходимости следует произвести перезагрузку ПК. Чтобы проконтролировать правильность распознавания операционной системой подключенного модуля необходимо выполнить:

Start -> Settings -> Control Panel -> System -> Devices -> Device Manager

(Старт -> Настройка -> Панель управления -> Система -> Оборудование -> Диспетчер устройств).

В разделе «Контроллеры универсальной последовательной шины USB» должно отображаться устройство под названием «uDAQ2 Driver».

Обзор аппаратной части

Микросистемы m-DAQ12 и m-DAQ14 содержат восемь каналов аналогового ввода $A_{IN0} \dots A_{IN7}$. Диапазон входных напряжений устанавливается программно для каждого канала и составляет ± 10 В, ± 5 В, ± 2.5 В, ± 1.25 В. Аналого-цифровой тракт содержит 8-канальный коммутатор выбора опрашиваемого канала, коммутатор режима подключения, инструментальный усилитель с изменяемым коэффициентом усиления, собственно сам 12-ти либо 14-ти разрядный АЦП. Буфер FIFO и автомат управления, реализованы на основе микроконтроллера.

Гарантированное максимальное значение частоты дискретизации в одноканальном режиме составляет 150 кГц, в многоканальном режиме – 100 кГц. Запуск АЦП осуществляется от внутреннего программируемого генератора, а начало процесса оцифровки может быть синхронизировано с внешним событием.

Аналоговый тракт АЦП выполняет функцию входной коммутации каналов и поканальной установки коэффициента усиления. Каждый входной канал имеет индивидуальные настройки: режим коммутации и индивидуальный коэффициент усиления аналогового тракта (который соответствует четырем диапазонам входных напряжений).

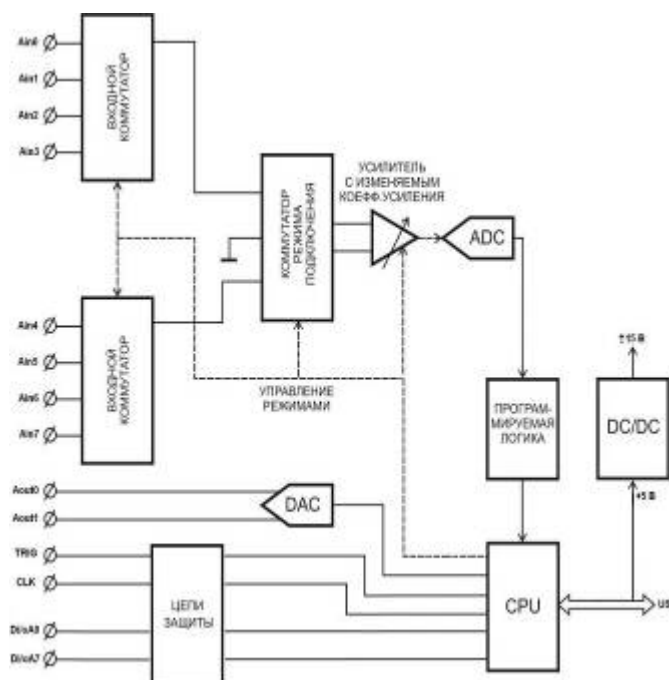
Программируемая входная коммутация позволяет гибко настроить модуль на необходимый режим, определяемый способом подключения входных сигналов:

1. Дифференциальный режим подключения – от 1 до 4 каналов.
2. Однопроводный режим подключения – от 1 до 8 каналов.
3. Внутреннее соединение входа АЦП с аналоговой землей модуля.

В первом режиме модуль позволяет опросить до 4 дифференциальных каналов. Оцифрованный отсчет представляет собой разность сигналов ($A_{IN\ 0..3} - A_{IN\ 4..7}$) относительно земли (A_{GND}).

Во втором режиме модуль позволяет опросить до 8 каналов. Оцифрованный отсчет представляет собой уровень сигнала ($A_{IN\ 0..7}$) относительно земли (A_{GND}).

В третьем режиме коммутатор позволяет подать на вход АЦП (через усилитель с изменяемым коэффициентом усиления) нулевое напряжение и измерить фактическое входное напряжение смещения нуля АЦП модуля. В этом режиме коммутатор отключает входные сигнальные цепи всех каналов.



АППАРАТНАЯ ЧАСТЬ

Таким образом, устанавливая необходимый режим подключения, количество опрашиваемых входных каналов может быть гибко настроено от 1 до 8. Соответственно частота запуска АЦП делится между опрашиваемыми каналами в соответствии с их количеством и порядком (кратностью) опроса.

Аналоговый вывод A_{OUT0} и A_{OUT1} реализован в m-DAQ12 и m-DAQ14 в виде двух независимых каналов ЦАП с программно устанавливаемым выходным диапазоном напряжений ± 5 В, ± 10 В, ± 10.8 В, $0 \dots 5$ В, $0 \dots 10$ В и $0 \dots 10.8$ В и током нагрузки 5 мА.

Дискретный В/В (ТТЛ) в микросистемах представлен одним 8-битовым портом $D_{I/O}A0 \dots D_{I/O}A7$, причем каждая линия $D_{I/O}$ может быть сконфигурирована индивидуально на ввод или вывод. Линия CLK является входом (ТТЛ) 16-разрядного таймера-счетчика внешних событий, а линия TRIG может использоваться для запуска оцифровки данных трактом АЦП.

Каждая из линий $D_{I/O}A0 \dots D_{I/O}A7$, TRIG, CLK содержит встроенный токоограничивающий резистор сопротивлением 430 Ом и встроенный «подтягивающий» (Pull-Up) резистор сопротивлением 10 кОм к линии +5 В.

Питание аналоговых цепей микросистем осуществляется напряжением ± 15 В от встроенного DC/DC-преобразователя. Напряжение ± 15 В выведено на входной разъем устройства через самовосстанавливаемые предохранители и может быть использовано для питания внешних слаботочных устройств, например, нормализаторов сигналов датчиков. На контактах входного разъема присутствует также и напряжение +5 В шины USB, выведенное через плавкий предохранитель в интегральном исполнении.



Аналоговый и дискретный тракты не имеют гальванической развязки от шины USB!

Функциональные возможности

Микросистемы сбора данных m-DAQ12 и m-DAQ14, как комплекс аппаратных и программных средств, обеспечивают:

- синхронный ввод данных с одного или нескольких аналоговых входов в массив, заданный пользователем;
- асинхронный и синхронный ввод (вывод) данных с (на) линий дискретного В/В $D_{I/O} A0..A7$;
- побитовую асинхронную установку и чтение линий дискретного В/В портов $D_{I/O} A0..A7$;
- подсчет импульсов импульсного сигнала на входе CLK.

Функциональные возможности микросистем реализуются системой команд:

- установка частоты дискретизации АЦП;
- установка номеров опрашиваемых каналов АЦП, режимов их подключения и входных диапазонов;
- чтение массива данных с АЦП;
- чтение массивов данных с АЦП в непрерывном режиме;
- чтение состояния линий порта дискретного В/В;
- установка данных в порт дискретного В/В;
- установка частоты чтения/записи данных с (на) линий порта дискретного В/В;
- чтение массива данных с линий дискретного В/В;
- запись массива данных в порт дискретного В/В;
- запуск счетчика внешних событий;
- останов счетчика внешних событий;
- чтение результатов счетчика внешних событий;
- установка выходных диапазонов ЦАП (аналоговых выводов $A_{OUT} 0$ и $A_{OUT} 1$);
- вывод данных на ЦАП;
- конфигурация и назначение линий портов дискретного В/В.

Максимальное количество устройств m-DAQ12 и m-DAQ14, которые можно одновременно подключить к ПК -16 шт.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

Общие технические характеристики

| | |
|--|--|
| Напряжение питания | От шины USB 2.0 с возможностью работы в режиме HighSpeed (480Mb/s) |
| Ток потребления без нагрузки на ЦАП и линиях дискретного В/В, не более | 250 мА |
| Рабочая температура | от 5 °С до +40 °С |
| Относительная влажность воздуха | 5~90 % без конденсации влаги |
| Температура хранения | от -30 °С до +70 °С |
| Тип разъема для подключения аналоговых и дискретных сигналов | DB-25F (базовая модель); IDC-26 (ОЕМ-версия) |
| Материал корпуса | Пластик ABC |
| Габаритные размеры | 60x100x28 мм |
| Масса, не более | 0,35 кг |

Технические характеристики аналоговых каналов АЦП m-DAQ12

| | |
|---|---|
| Количество аналоговых входов | 4 дифференциальных канала или 8 однопроводных каналов с общей землей |
| Входной диапазон | $\pm 10 \text{ В}$, $\pm 5 \text{ В}$, $\pm 2.5 \text{ В}$, $\pm 1.25 \text{ В}$ |
| Предельное входное напряжение | $\pm 15 \text{ В}$ относительно A_{GND} |
| Предельное входное напряжение при отключенном питании модуля | $\pm 10 \text{ В}$ относительно A_{GND} |
| Разрядность АЦП | 12 бит |
| Дифференциальная нелинейность преобразования | Макс. -1 ..+1.5 МЗР |
| Интегральная нелинейность преобразования | Макс. $\pm 1.5 \text{ МЗР}$ |
| Частота дискретизации (многоканальный режим / одноканальный режим) | 65 Гц .. 100 кГц / 65 Гц .. 150 кГц |
| Межканальное прохождение синусоидального сигнала 1 кГц на диапазоне $\pm 10 \text{ В}$, и внутреннем сопротивлении источников сигналов 50 Ом | 98 дБ при частоте преобразования АЦП 10,0 кГц 93 дБ при частоте преобразования АЦП 100,0 кГц |
| Подавление синфазной составляющей при дифференциальном подключении | 92 дБ для диапазона $\pm 10 \text{ В}$, 96 дБ для диапазона $\pm 5 \text{ В}$, 100 дБ для диапазонов $\pm 2.5 \text{ В}$, $\pm 1.25 \text{ В}$ |

Технические характеристики аналоговых каналов АЦП m-DAQ14

| | |
|--|---|
| Количество аналоговых входов | 4 дифференциальных канала или 8 однопроводных каналов с общей землей |
| Входной диапазон | $\pm 10 \text{ В}$, $\pm 5 \text{ В}$, $\pm 2.5 \text{ В}$, $\pm 1.25 \text{ В}$ |
| Предельное входное напряжение | $\pm 15 \text{ В}$ относительно A_{GND} |
| Предельное входное напряжение при отключенном питании модуля | $\pm 10 \text{ В}$ относительно A_{GND} |
| Разрядность АЦП | 14 бит |
| Дифференциальная нелинейность преобразования | Макс. -1 ..+1.5 МЗР |

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

| | |
|--|--|
| Интегральная нелинейность преобразования | Макс. ± 1.5 МЗР |
| Частота дискретизации (многоканальный режим / одноканальный режим) | 65 Гц .. 100 кГц / 65 Гц .. 150 кГц |
| Межканальное прохождение синусоидального сигнала 1 кГц на диапазоне ± 10 В, и внутреннем сопротивлении источников сигналов 50 Ом | 95 дБ при частоте преобразования АЦП 10,0 кГц 80 дБ при частоте преобразования АЦП 100,0 кГц |
| Подавление синфазной составляющей при дифференциальном подключении | 92 дБ для диапазона ± 10 В, 96 дБ для диапазона ± 5 В, 100 дБ для диапазонов ± 2.5 В, ± 1.25 В |

Технические характеристики аналоговых каналов ЦАП

| | |
|-------------------------------|---|
| Количество аналоговых выходов | 2 |
| Разрядность ЦАП | 12 бит |
| Время установления | 10 мкс |
| Выходной диапазон | 0..5 В, 0..10 В, 0..10.8 В, ± 5 В, ± 10 В, ± 10.8 В |
| Сопротивление нагрузки | не менее 2 кОм |
| Максимальный ток нагрузки | ± 5 мА |

Технические характеристики каналов дискретного ввода/вывода

| | |
|--|--------------------------------------|
| Общее число линий В/В | 8 |
| Тип логики | TTL (3.3 В) |
| Выходной ток лог. «1» (относительно земли D_{GND}) | 7 мА |
| Выходной ток лог. «0» (относительно питания 5 В) | 10 мА |
| Напряжение лог. «1» (конфигурация как выход) | 3.3 В |
| Напряжение лог. «1» (конфигурация как вход) | 3.3 – 5.0 В |
| Напряжение лог. «0» | не более 0.1 В |
| Выходное сопротивление канала (конфигурация как выход) | 430 Ом |
| Входное сопротивление канала (конфигурация как вход) | 10.5 кОм (внутренний Pull-Up к +5 В) |
| Частота синхронного В/В | 50 кГц .. 5 МГц |

Технические характеристики каналов CLK, TRIG, $\pm V$, +5V

| | |
|--|---|
| Тип логики по каналам CLK, TRIG | TTL (3.3 В) |
| Ток, напряжение, сопротивление каналов CLK, TRIG | Аналогично каналам дискретного ввода/вывода |
| Емкость счетчика событий | 65535 |
| Выходы для питания внешних цепей $\pm V$, +5V | ± 15 В/30 мА, +5 В/100 мА (суммарная мощность по выходам +5 В и ± 15 В не должна превышать 1 Вт) |

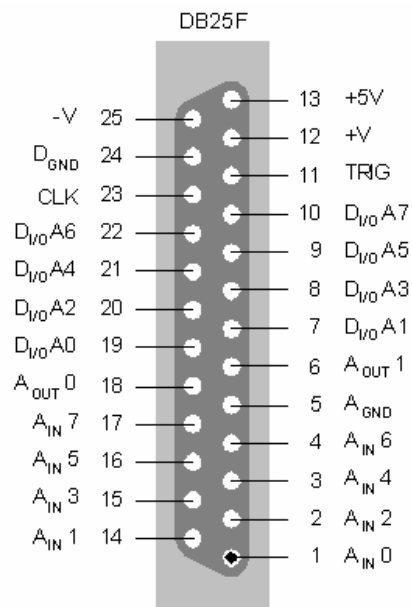
ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ**Информация для заказа**

| | |
|-------------------------------|--|
| <i>m-DAQ12</i> | АЦП 12 бит, 4/8 кан., 100 кГц, ± 10 В, ± 5 В, ± 2.5 В, ± 1.25 В, 8 TTL (3.3 В) вх./вых., интерфейс USB 2.0. HighSpeed (480Mb/s). |
| <i>m-DAQ12/DAC</i> | АЦП 12 бит, 4/8 кан., 100 кГц, ± 10 В, ± 5 В, ± 2.5 В, ± 1.25 В, 8 TTL (3.3 В) вх./вых., интерфейс USB 2.0. HighSpeed (480Mb/s). + ЦАП 12 бит, 2 канала, 0..5 В, 0..10 В, ± 5 В, ± 10 В. |
| <i>m-DAQ12/OEM</i> | АЦП 12 бит, 4/8 кан., 100 кГц, ± 10 В, ± 5 В, ± 2.5 В, ± 1.25 В, 8 TTL (3.3 В) вх./вых., интерфейс USB 2.0. HighSpeed (480Mb/s). Поставка микросистемы сбора данных без корпуса. |
| <i>m-DAQ12/DAC/OEM</i> | АЦП 12 бит, 4/8 кан., 100 кГц, ± 10 В, ± 5 В, ± 2.5 В, ± 1.25 В, 8 TTL (3.3 В) вх./вых., интерфейс USB 2.0. HighSpeed (480Mb/s). + ЦАП 12 бит, 2 канала, 0..5 В, 0..10 В, ± 5 В, ± 10 В. Поставка микросистемы сбора данных без корпуса. |

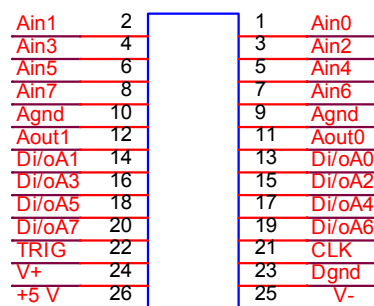
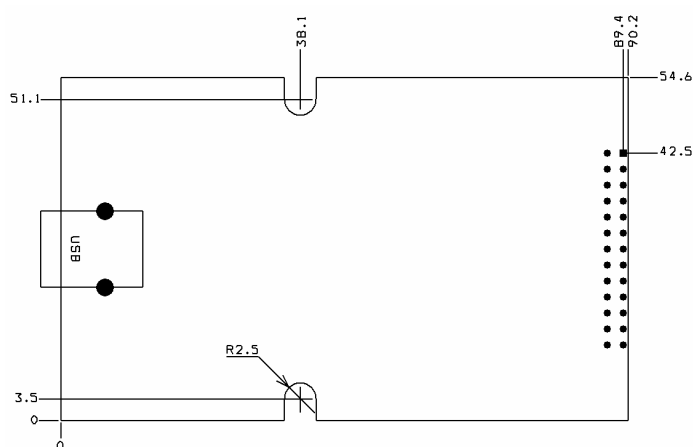
| | |
|-------------------------------|--|
| <i>m-DAQ14</i> | АЦП 14 бит, 4/8 кан., 100 кГц, ± 10 В, ± 5 В, ± 2.5 В, ± 1.25 В, 8 TTL (3.3 В) вх./вых., интерфейс USB 2.0. HighSpeed (480Mb/s). |
| <i>m-DAQ14/DAC</i> | АЦП 14 бит, 4/8 кан., 100 кГц, ± 10 В, ± 5 В, ± 2.5 В, ± 1.25 В, 8 TTL (3.3 В) вх./вых., интерфейс USB 2.0. HighSpeed (480Mb/s). + ЦАП 12 бит, 2 канала, 0..5 В, 0..10 В, ± 5 В, ± 10 В. |
| <i>m-DAQ14/OEM</i> | АЦП 14 бит, 4/8 кан., 100 кГц, ± 10 В, ± 5 В, ± 2.5 В, ± 1.25 В, 8 TTL (3.3 В) вх./вых., интерфейс USB 2.0. HighSpeed (480Mb/s). Поставка микросистемы сбора данных без корпуса. |
| <i>m-DAQ14/DAC/OEM</i> | АЦП 14 бит, 4/8 кан., 100 кГц, ± 10 В, ± 5 В, ± 2.5 В, ± 1.25 В, 8 TTL (3.3 В) вх./вых., интерфейс USB 2.0. HighSpeed (480Mb/s). + ЦАП 12 бит, 2 канала, 0..5 В, 0..10 В, ± 5 В, ± 10 В. Поставка микросистемы сбора данных без корпуса. |

ПОДКЛЮЧЕНИЕ СИГНАЛОВ

В m-DAQ12 и m-DAQ14 подключение источников/приемников сигналов выполняется через соединитель типа DB25F:



В версиях m-DAQ12/OEM и m-DAQ14/OEM подключение источников/приемников сигналов выполняется через соединитель типа IDC26:



В обозначениях контактов соединителей: A_{GND} – общий аналоговых сигналов, D_{GND} – общий дискретных сигналов.

ПОДКЛЮЧЕНИЕ СИГНАЛОВ

Подключение сигналов и распайка разъема возлагаются на пользователя. Монтаж сигнальных цепей с подключением источников сигнала, датчиков и т.п. к модулю должен выполнять специалист соответствующей квалификации.

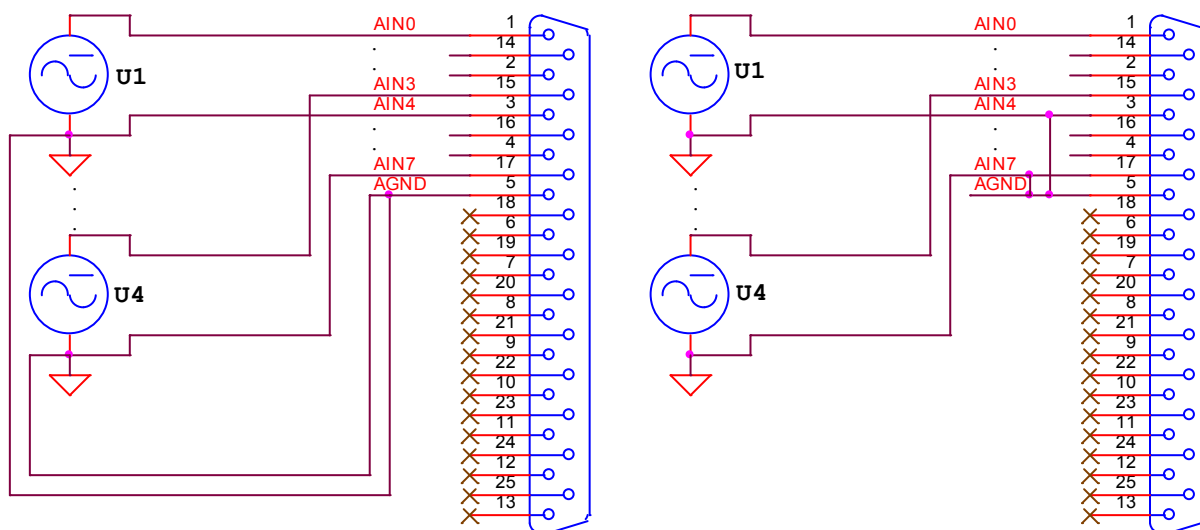
Подключения, значения уровней токов и напряжений при которых превышают предельно допустимые значения, влекут за собой ухудшение параметров модуля или выход из строя компьютера, модуля, или присоединенного оборудования.

Цепи A_{GND} и D_{GND} , выведенные на контакты разъема, связаны между собой внутри модуля, а также с общим проводом USB и корпусом компьютера. При подключении микросистем к внешним цепям следует помнить, что правильным подключением является то, которое не приводит к протеканию сквозных токов по цепям $D_{GND}-A_{GND}$, D_{GND} –корпус компьютера или $D_{GND}-A_{GND}$ –корпус компьютера. Наличие вышеуказанных токов может ухудшить соотношение сигнал-шум в каналах модуля, вызвать неустойчивую работу шины USB, а при сквозном токе более 200 мА вызвать неисправность модуля или USB-порта компьютера. Если же такие токи при использовании модуля по какой-либо причине неизбежны, то следует принять меры по их минимизации и подавлению их высокочастотной составляющей.

Примеры подключения к аналоговым каналам

Рассмотрим типовые схемы подключения сигналов.

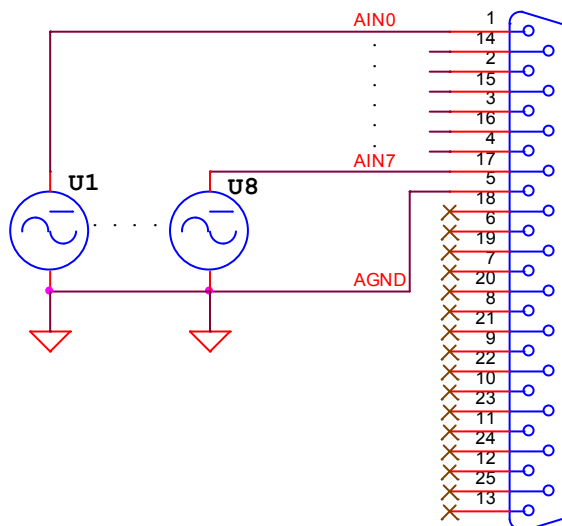
Подключение источников напряжения в режиме 4-х дифференциальных каналов.



Пример подключения источников напряжения

ПОДКЛЮЧЕНИЕ СИГНАЛОВ

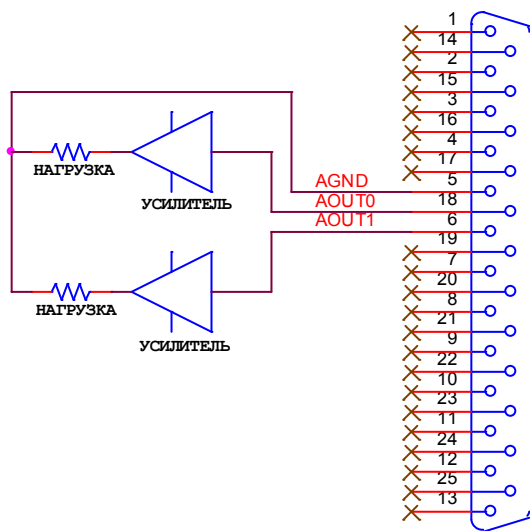
Подключение источников напряжения в режиме 8-и однопроводных каналов.



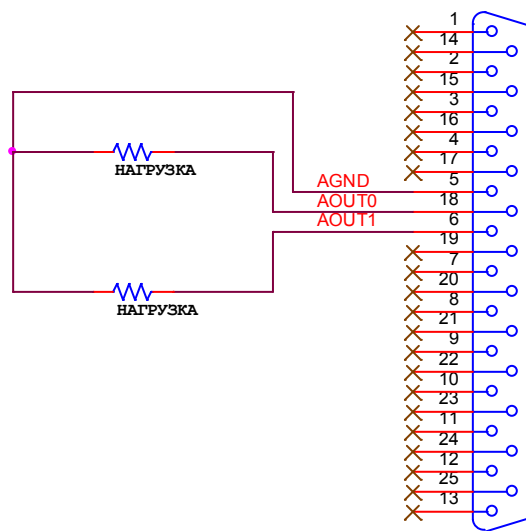
Пример подключения источников напряжения

Подключение приемников сигналов к выходам ЦАП

а) Пример подключения «мощной» нагрузки.



б) Пример подключения маломощной нагрузки.



Пример подключения приемников сигналов



Как показывает опыт, более 80 % проблем, возникающих при эксплуатации модуля, связаны с неправильным подключением источников сигналов.

ПОДКЛЮЧЕНИЕ СИГНАЛОВ

При дифференциальной схеме подключения сигнала измеряется разность напряжений между двумя входами ($A_{IN\ 0..3}$ - $A_{IN\ 4..7}$) канала относительно A_{GND} . При таком подключении обеспечивается оптимальное подавление помех от внешней среды, наводимых на соединительные провода. Однопроводная схема подключения более чувствительна к фактору внешних помех.

Модули m-DAQ12 и m-DAQ14 не имеет внутренней гальванической развязки, поэтому цепь земли (A_{GND} и D_{GND}) имеет внутреннее соединение с общим проводом USB и, следовательно, с заземлением ПК.

Следует помнить, что для корректной работы входного тракта модуля необходимо, чтобы потенциал каждого входа $A_{IN\ 0..3}$ и $A_{IN\ 4..7}$ относительно земли A_{GND} – т.н. синфазное напряжение – не превышал установленного входного диапазона.



Неподключенные аналоговые входы необходимо заземлять (т.е. подсоединить к A_{GND}).

Следует также учитывать, что полоса пропускания входного тракта модуля выше максимальной частоты дискретизации АЦП. Для достижения спектральной верности преобразования сигнала следует ограничить полосу входного сигнала в соответствии с критерием Найквиста. Это означает, что необходимо ограничить высокочастотную составляющую сигнала до приемлемого уровня шумов, начиная с частоты $0,5 \cdot F_s$ (где F_s – частота дискретизации АЦП) и выше. В противном случае все шумы и помехи, имеющие частоту выше $0,5 \cdot F_s$, будут накладываться на полезный сигнал в рабочей полосе частот и не смогут быть отделены от него при последующей обработке.

В случае многоканального ввода сигналов приходится учитывать наличие входной емкости коммутаторов аналогового тракта $C_{in} \sim 100$ пФ.

Паразитное межканальное прохождение не превысит погрешности преобразования АЦП если выполняется следующее условие:

$$R_s \cdot C_{in} \leq 0,1 \cdot (1 / F_s),$$

где R_s – выходное сопротивление источника сигнала. Для случаев большого уровня входного сигнала (вне установленного диапазона модуля), а также в случае коммутации каналов с разным установленным входным диапазоном, следует применять более жесткий критерий, учитывающий увеличение межканального прохождения в этих случаях.



При отключенном интерфейсе USB или отключенном питании ПК уменьшаются предельно допустимые входные напряжения модулей.

| |
|--------------------------------|
| ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ |
|--------------------------------|

Программное обеспечение m-DAQ12 и m-DAQ14 содержит USB-драйвер, DLL-библиотеку, библиотеку функций и пример (осциллограф-регистратор) в среде графического программирования LabVIEW.

Шина USB предоставляет возможность работать с периферийными устройствами в режиме Plug&Play. Это означает, что стандартом USB предусмотрено подключение устройства к работающему компьютеру, автоматическое его распознавание сразу же после подключения и последующая загрузка операционной системой соответствующих данному устройству драйверов.

Интерфейс USB компьютера должен соответствовать спецификации USB 2.0 и поддерживать режим HighSpeed (480Mb/s).

При первом подключении модуля к компьютеру операционная система запросит файлы драйвера для модуля. При этом подразумевается, что на компьютере уже установлена операционная система, способная корректно поддерживать функционирование USB-шины. Пользователю необходимо указать расположение файлов *mdaq2ldr.inf* и *mdaq2.inf* (последовательно). В случае успешной инициализации информация о драйвере будет занесена в реестр Windows, и при повторных сеансах работы модуль будет инициализироваться автоматически. При необходимости следует выполнить перезагрузку компьютера.

| | |
|----------------|-----------------------------|
| DRIVER\ | mdaq2.spt |
| | cyusb.sys |
| | mdaq2.inf |
| | mdaq2ldr.inf |
| DLL\ | mdaq12.dll |
| | mdaq12.lib |
| | mdaq12.h |
| DEMO\ | console_test.exe |
| | mdaq2dll.dll |
| | console_src\ |
| LV_LIBRARY\ | m-DAQ12 library.llb |
| | mdaq12.dll |
| C++ 2008 SP1\ | vcredist_x86.exe |
| LV_RUN-TIME\ | LabVIEW821RuntimeEngine.exe |
| LV_INSTRUMENT\ | Oscilloscope-logger\ |

DLL-библиотека mDAQ12.dll содержит набор функций, с помощью которых можно реализовывать различные алгоритмы ввода/вывода. Для вызова функций DLL-библиотеки из приложения пользователя необходимо выполнить следующее:

- создать проект в какой либо из сред разработки;
- подключить к проекту файлы mdaq2.lib и mdaq2.h;
- создать и добавить в проект файл с исходным текстом будущей программы;
- с помощью функции GetDllVersion(), желательно сравнить версию используемой dll-библиотеки с версией текущего программного обеспечения;
- вызвать функцию GetUsbDeviceCount(), чтобы определить количество подключенных m-DAQ12 и m-DAQ14;
- вызвать функцию OpenDevice(), чтобы открыть требуемое устройство;
- вызвать другие функции, которые необходимы для работы.

Все интерфейсные функции не обеспечивают “потокобезопасную” работу dll-библиотеки. Поэтому, во избежание недоразумений, в многопоточных приложениях пользователь должен сам организовывать, если необходимо, корректную синхронизацию вызовов интерфейсных функций в различных потоках (используя, например, критические участки, мьютексы и т.д.).

В файл библиотеки mdaq12.dll включена информация о текущей версии dll. Для получения сведений о текущей версии DLL библиотеки следует использовать функцию GetDllVersion(). Если версии совпадают, то необходимо вызвать функции GetUsbDeviceCount() и OpenDevice(), чтобы открыть модуль. После этого можно управлять всей доступной периферией на модуле с помощью соответствующих интерфейсных функций DLL-библиотеки.



В случае получения следующей ошибки ОС при работе с dll-библиотекой: *«Приложение не было запущено, поскольку оно некорректно настроено. Повторная установка приложения может решить данную проблему»*, необходимо предварительно установить распространяемый пакет Microsoft Visual C++ 2008 SP1 Redistributable Package из папки C++ 2008 SP1\ поставляемого диска.

Используемые термины и форматы данных

| | |
|--------------|--|
| DevN | номер открываемого устройства |
| id | указатель на устройство |
| ParN | номер возвращаемого параметра |
| *ParString | указатель на буфер, принимающий заданную информацию |
| *Name | указатель на строку, содержащую задаваемое имя |
| On | булева переменная для управления светодиодом |
| freq | частота дискретизации |
| Count | количество опрашиваемых каналов |
| *_array | указатель на массив кодов каналов |
| Chan | номер опрашиваемого канала |
| Amp | код входного диапазона АЦП |
| Mode | код режима подключения |
| *OutVal | указатель на переменную, принимающую код канала |
| *inBuf | указатель на буфер, принимающий массив данных |
| Length | размер массива (в отсчетах) |
| Timeout | таймаут ожидания окончания работы функции |
| WaitComplete | булева переменная для задания режима ожидания данных |
| dac | номер канала ЦАП |
| diap | код диапазона ЦАП |
| val | преобразуемый код ЦАП |
| enable | булева переменная для запуска либо останова счетчика |
| *data | указатель на переменную, принимающую байт данных |
| data | байт данных |
| DOorDI | переменная для задания цифрового ввода либо вывода |
| *OutBuf | указатель на массив данных, заданных пользователем |

Функции DLL-библиотеки

| | |
|---|--|
| GetDllVersion() | получение версии DLL-библиотеки |
| GetUsbDeviceCount() | получение количества подключенных устройств |
| OpenDevice(int DevN = -1) | открытие требуемого устройства и получение указателя на него |
| CloseDevice(unsigned char id = -1) | закрытие устройства |
| GetDeviceInfo(unsigned char id, unsigned char ParN, char *ParString); | получение информации об устройстве (наименование, серийный номер) |
| isDeviceOpen(unsigned char id) | проверка открытия устройства |
| SetDeviceUserName(char *Name) | установка наименования устройства |
| *ErrorToText(signed char ErrN) | преобразование кода ошибки в соответствующий текст |
| AltErrorToText(signed char ErrN, char *ErrStr) | преобразование кода ошибки в соответствующий текст |
| LedOnOff(unsigned char id, bool On) | управление установленным на модуле светодиодным индикатором |
| SetADCFreq(unsigned char id, unsigned long freq) | установка частоты дискретизации АЦП |
| SetADCChannels(unsigned char id, unsigned char *_array, unsigned char Count) | установка параметров опроса каналов АЦП |
| ConvertADCChannelToVal(unsigned char Chan, unsigned char Mode, unsigned char Amp, unsigned char *OutVal) | преобразование номера канала, режима подключения и входного диапазона АЦП в код канала, используемый при вызове SetADCChannels |
| ReadADCFrame(unsigned char id, signed short *inBuf, LONG Length, unsigned long Timeout = 1000, bool WaitComplete = true, unsigned char ExtTrig = 0) | чтение массива данных с АЦП |
| ReadADCFrameMaxFreq(unsigned char id, unsigned char Chan, signed short *inBuf, LONG Length, unsigned long Timeout = 1000, bool WaitComplete = true, unsigned char ExtTrig = 0) | чтение массива данных с одного канала АЦП с максимальной скоростью |
| isADCXferComplete(unsigned char id) | проверка, закончена ли передача данных от устройства к PC при выполнении функций ReadADCFrame, ReadADCFrameMaxFreq либо ContinuousReadADCFrame |
| StartContinuousReadADC(unsigned char id) | запуск АЦП в режиме непрерывного чтения |
| ContinuousReadADCFrame(unsigned char id, signed short *inBuf, LONG Length) | чтение массива данных с АЦП в режиме непрерывного чтения |
| ContinuousReadADCFrame_OVL(unsigned char id, signed short *inBuf, LONG Length, unsigned int OvlN, bool WaitComplete = true, unsigned long Timeout = 1000); | чтение массива данных с АЦП в режиме непрерывного чтения (альтернативная ContinuousReadADCFrame функция) |
| isADCXferComplete_OVL(unsigned char id, unsigned int OvlN) | проверка, закончена ли передача данных от устройства к PC при выполнении функции ContinuousReadADCFrame_OVL |
| StopContinuousReadADC(unsigned char id) | останов режима непрерывного чтения АЦП |

| | |
|---|--|
| SetDacRange(unsigned char id, unsigned char dac, unsigned char diap) | установка выходного диапазона канала ЦАП |
| SetDacValue(unsigned char id, unsigned char dac, int val) | установка напряжения на канале ЦАП |
| PowerUpDac(unsigned char id, unsigned char dac) | включение канала ЦАП после срабатывания защиты от перегрузки по току |
| ClearDACs(unsigned char id) | сброс каналов ЦАП |
| EnableCounter(unsigned char id, BOOL enable) | запуск либо останов счетчика событий |
| ReadCounter(unsigned char id, unsigned int *data) | чтение значений счетчика событий |
| ClearCounter(unsigned char id) | сброс счетчика событий |
| ConfigureDigitalPort(unsigned char data) | конфигурирование линий порта дискретного В/В для однократного В/В |
| ReadDIValue(unsigned char id, unsigned char *data) | однократный ввод данных с порта дискретного В/В |
| SetDOValue(unsigned char id, unsigned char data) | однократный вывод данных в порт дискретного В/В |
| SetFastDIOFreq(unsigned char id, unsigned long freq, unsigned char DOorDI) | установка частоты опроса порта дискретного В/В |
| FastReadDI(unsigned char id, unsigned char *InBuf, LONG Length, unsigned long Timeout, bool WaitComplete = true) | чтение массива данных с порта дискретного В/В |
| FastWriteDO(unsigned char id, unsigned char *OutBuf, LONG Length, unsigned long Timeout, bool WaitComplete = true) | вывод массива данных в порт дискретного В/В |
| isReadDIXferComplete(unsigned char id, unsigned char *InBuf) | проверка, закончена ли передача данных от устройства к PC при выполнении функции FastReadDI |
| isWriteDOXferComplete(unsigned char id) | проверка, закончена ли передача данных от PC к устройству при выполнении функции FastWriteDO |

Описание кодов ошибок

| Код ошибки | Описание ошибки |
|-------------------|--|
| -1 | Устройство не подключено к USB |
| -2 | Некорректный указатель на открываемое устройство |
| -3 | Более 16-ти подключенных к PC устройств |
| -4 | Ошибка открытия |
| -5 | В устройство не загружена прошивка |
| -6 | Внутренняя ошибка прошивки устройства |
| -7 | Устройство не открыто |
| -8 | Устройство уже открыто |
| -9 | К интерфейсу USB подключено отличное от m-DAQ12 и m-DAQ14 устройство |

Функции общего назначения

USHORT __stdcall GetDllVersion()

Получение версии DLL библиотеки. Возвращает версию DLL-библиотеки - двухбайтное число, обозначающее номер версии, передаваемые параметры отсутствуют.

USHORT __stdcall GetUsbDeviceCount()

Получение количества подключенных к шине USB устройств.

int __stdcall OpenDevice(int DevN = -1)

Открытие устройства, подключенного к интерфейсу USB компьютера, и получение указателя на него. Подсоединенный к компьютеру модуль m-DAQ12 либо m-DAQ14 можно рассматривать как устройство, подключенное к некоему виртуальному слоту с индивидуальным номером. Основное назначение функции состоит в том, чтобы определить, что именно m-DAQ12/m-DAQ14 находится в заданном виртуальном слоте. Если выполнение функции OpenDevice прошло успешно для заданного виртуального слота, то можно переходить к последующему управлению работой модуля. **int DevN** – номер модуля, который необходимо открыть (нумерация начинается с 0 в порядке подключения устройств к USB). При задании DevN = -1 функция выполняет открытие всех подключенных устройств и возвращает их количество, а индексы нумеруются, начиная с 0. Отрицательное возвращаемое значение функции свидетельствует об ошибке выполнения и одновременно является кодом ошибки.

bool __stdcall CloseDevice(unsigned char id = -1)

Закрытие устройства – закрывается дескриптор требуемого устройства. Здесь и далее unsigned char id – указатель на данное устройство, с помощью которого производится адресация команд. Здесь и далее возвращаемое значение: TRUE – функция выполнена успешно, FALSE – ошибка при выполнении функции.

bool __stdcall isDeviceOpen(unsigned char id)

Проверка открытия устройства. Если устройство заданным id открыто, возвращаемое значение функции – TRUE.

bool __stdcall SetDeviceUserName(char *Name)

Установка наименования устройства. Передаваемый параметр char *Name - указатель на строку, содержащую задаваемое имя (не более 16 символов).

char __stdcall *ErrorToText(signed char ErrN)

Преобразование кода ошибки в соответствующий текст. Передаваемый параметр signed char ErrN – код ошибки.

AltErrorToText(signed char ErrN, char *ErrStr)

Преобразование кода ошибки в соответствующий текст. Передаваемый параметр signed char ErrN – код ошибки. char *ErrStr – указатель на буфер, в который будет помещена строка ошибки

bool __stdcall LedOnOff(unsigned char id, bool On)

Управление установленным на модуле светодиодным индикатором. Передаваемый параметр bool On служит для включения (в случае On = TRUE) либо выключения светодиода.

bool __stdcall GetDeviceInfo(unsigned char id, unsigned char ParN, char *ParString)

Получение информации об устройстве.

Передаваемые параметры: unsigned char id, unsigned char ParN – номер возвращаемого параметра (соответствие номеров и параметров представлено в соотв. таблице ниже) char *ParString – указатель на буфер, принимающий заданную информацию.

| Номер параметра ParN | Параметр |
|----------------------|--|
| 1 | Наименование устройства, заданное пользователем |
| 2 | Серийный номер устройства |
| 3 | Наименование устройства, записанное в inf-файле |
| 4 | Код производителя, записанный в inf-файле |
| 5 | Код устройства, записанный в inf-файле |
| 6 | Версия микропрограммы устройства |
| 10 | Скорость подключения по шине USB, «HS» (HighSpeed) либо «FS» (FullSpeed) |
| 11 | Разрядность АЦП, бит |

Функции работы с АЦП

bool __stdcall SetADCFreq(unsigned char id, unsigned long freq)

Установка частоты дискретизации. Данную функцию необходимо использовать перед началом сбора данных. Значение частоты задается с помощью внутреннего таймера в диапазоне 65 Гц..100 кГц и вычисляется по формуле: $F=4 \cdot 10^6/k$, где k – целое число 40..61538. Передаваемые параметры: unsigned char id, unsigned long freq – частота дискретизации (65..100000).

bool __stdcall SetADCChannels(unsigned char id, unsigned char *_array, unsigned char Count)

Установка параметров опроса каналов АЦП. Передаваемые параметры: unsigned char id, unsigned char *_array – указатель на массив кодов каналов, размером 8 байт (размер кода каждого канала – 1 байт). unsigned char Count – количество опрашиваемых каналов. При чтении данных с АЦП каналы опрашиваются циклически в соответствии с информацией в массиве кодов каналов. Код канала зависит от номера опрашиваемого канала, кода режима подключения и кода входного диапазона АЦП и вычисляется с использованием функции ConvertADCChannelToVal.

bool __stdcall ConvertADCChannelToVal(unsigned char Chan, unsigned char Mode, unsigned char Amp, unsigned char *OutVal)

Преобразование номера канала, режима подключения и входного диапазона АЦП в код канала, используемый при вызове SetADCChannels. Передаваемые параметры: unsigned char id – указатель на устройство, unsigned char Chan, unsigned char Mode, unsigned char Amp – номер опрашиваемого канала, код режима подключения, код входного диапазона АЦП соответственно (см. таблицу ниже), unsigned char &OutVal – указатель на переменную, принимающую код канала.

| Переменная | Описание | Диапазон устанавливаемых значений | Описание значений |
|------------|----------------------------|--|---|
| Chan | Номер опрашиваемого канала | для дифф. подключения – 0..3; для однопроводного подключения – 0..7 | Используемые входы для дифф. подключения: $Ch = 0 - A_{IN} 0$ и $A_{IN} 4$; $Ch = 1 - A_{IN} 1$ и $A_{IN} 5$; $Ch = 2 - A_{IN} 2$ и $A_{IN} 6$; $Ch = 3 - A_{IN} 3$ и $A_{IN} 7$; для однопроводного подключения: $Ch = i - A_{IN} i$. |
| Mode | Код режима подключения | 0..2 | 0 – режим с заземленным входом; 1 – дифф. подключение; 2 – однопроводное подключение |
| Amp | Код входного диапазона АЦП | 1..4 | 1 – 0..10 В; 2 – 0..5 В; 3 – 0..2,5 В; 4 – 0..1,25 В |

```
bool __stdcall ReadADCFrame(unsigned char id, signed short *inBuf, LONG Length, unsigned long Timeout = 1000, bool WaitComplete = true, unsigned char ExtTrig = 0)
```

Чтение данных с АЦП в массив, размер которого задается пользователем. В многоканальном режиме данные записываются в возвращаемый массив в соответствии с порядком опроса каналов. Передаваемые параметры: unsigned char id, signed short *inBuf – указатель на буфер, принимающий массив данных (размер массива должен быть кратным 512 и не должен превышать $2 \cdot 10^6$), LONG Length – размер массива в отсчетах, unsigned long Timeout – таймаут ожидания окончания чтения данных (в ms, по умолчанию равен 1000, при превышении установленного значения чтение АЦП прерывается), bool WaitComplete – режим ожидания данных, unsigned char ExtTrig – режим внешней синхронизации (по линии TRIG, 0 – отключить, 1 – запуск АЦП по фронту сигнала на TRIG, 2 – запуск по спаду). При WaitComplete = TRUE функция ожидает завершения передачи данных, в противном случае выход из функции происходит сразу после начала передачи и пользователь имеет возможность обрабатывать ранее полученные данные (по умолчанию TRUE). Для проверки окончания передачи данных (при WaitComplete = FALSE) необходимо использовать функцию isADCXferComplete.



Во время передачи данных с АЦП обращение к устройству (кроме isADCXferComplete) недопустимо.



Для корректной работы данной функции необходимо равенство переданного по *inBuf значения и размера массива, заданного в Length.



При включенной внешней синхронизации (ExtTrig = 1 либо 2) значение на входе Timeout игнорируется.

```
bool __stdcall ReadADCFrameMaxFreq(unsigned char id, unsigned char Chan, signed short *inBuf, LONG Length, unsigned long Timeout = 1000, bool WaitComplete = true, unsigned char ExtTrig = 0)
```

Чтение данных с одного канала АЦП с максимальной частотой (150 кГц). Передаваемые параметры: unsigned char id, unsigned char Chan – код канала, signed short *inBuf, LONG Length, unsigned long Timeout, bool WaitComplete, unsigned char ExtTrig – идентичные одноименным передаваемым параметрам функции ReadADCFrame.

```
bool __stdcall isADCXferComplete(unsigned char id)
```

Проверка, закончена ли передача данных от устройства к PC при выполнении функций **ReadADCFrame**, **ReadADCFrameMaxFreq** либо **ContinuousReadADCFrame**. Передаваемый параметр – unsigned char id. Возвращаемое значение: FALSE – передача данных не закончена, TRUE – передача данных закончена.

```
bool __stdcall StartContinuousReadADC(unsigned char id)
```

Запуск АЦП в режиме непрерывного чтения. Режим непрерывного чтения характеризуется непрерывной работой АЦП. Вследствие этого пропуски информации между несколькими последовательно полученными с АЦП массивами данных отсутствуют. Передаваемый параметр – unsigned char id.



Так как емкость внутреннего буфера устройства (FIFO) ограничена 512 словами, вызовы функции **ContinuousReadADCFrame** должны происходить с такой периодичностью, чтобы не допускать переполнения внутреннего буфера на выбранной частоте дискретизации АЦП.

```
bool __stdcall ContinuousReadADCFrame(unsigned char id, signed short *inBuf, LONG Length)
```

Передача массива данных с АЦП в режиме непрерывного чтения. Выход из функции происходит сразу после начала передачи и пользователь имеет возможность обрабатывать ранее полученные данные. Для проверки окончания передачи данных необходимо использовать функцию **isADCXferComplete**. Функция может быть вызвана многократно (последовательно), при этом пропуски информации между несколькими последовательно полученными с АЦП массивами данных будут отсутствовать. Передаваемые параметры: unsigned char id, signed short *inBuf – указатель на буфер, принимающий массив данных (размер массива должен быть кратным 512 и не должен превышать $2 \cdot 10^6$), LONG Length – размер массива в отсчетах.



Перед передачей данных с АЦП в режиме непрерывного чтения необходимо предварительно выполнить запуск АЦП в соответствующем режиме с помощью функции **StartContinuousReadADC**. После окончания передачи необходимо выполнить останов АЦП (вызвать **StopContinuousReadADC**)



Повторный запуск **ContinuousReadADCFrame** необходимо выполнять только после окончания передачи предыдущих данных (при **isADCXferComplete** = TRUE).

```
bool __stdcall ContinuousReadADCFrame_OVL(unsigned char id, signed short *inBuf, LONG Length, unsigned int OvIN, bool WaitComplete = true, unsigned long Timeout = 1000);
```

Передача массива данных с АЦП в режиме непрерывного чтения (альтернативная ContinuousReadADCFrame функция). Выход из функции происходит сразу после начала передачи и пользователь имеет возможность обрабатывать ранее полученные данные. Для проверки окончания передачи данных необходимо использовать функцию isADCXferComplete. Функция может быть вызвана многократно, при этом пропуски информации между несколькими последовательно полученными с АЦП массивами данных будут отсутствовать. Передаваемые параметры: unsigned char id, signed short *inBuf, LONG Length, unsigned long Timeout, bool WaitComplete, unsigned char ExtTrig – идентичные одноименным передаваемым параметрам функции ContinuousReadADCFrame(). unsigned int OvIN – условный номер (0..1023) для функции в массиве ее очередности (также используется в функции isADCXferComplete_OVL()).



Перед передачей данных с АЦП в режиме непрерывного чтения необходимо предварительно выполнить запуск АЦП в соответствующем режиме с помощью функции StartContinuousReadADC. После окончания передачи необходимо выполнить останов АЦП (вызвать StopContinuousReadADC).



Последующую функцию ContinuousReadADCFrame_OVL можно вызвать сразу (с измененным значением OvIN), без ожидания окончания приема данных для ранее вызванной функции ContinuousReadADCFrame_OVL (при условии, что ее параметр WaitComplete = FALSE).

```
bool __stdcall isADCXferComplete_OVL(unsigned char id, unsigned int OvIN)
```

Проверка, закончена ли передача данных от устройства к РС при выполнении функции ContinuousReadADCFrame_OVL. Передаваемый параметр – unsigned char id. Возвращаемое значение: FALSE – передача данных не закончена, TRUE – передача данных закончена.

```
bool __stdcall StopContinuousReadADC(unsigned char id)
```

Останов режима непрерывного чтения АЦП. Передаваемый параметр – unsigned char id.

Функции работы с ЦАП

bool __stdcall SetDacRange(unsigned char id, unsigned char dac, unsigned char diap)

Установка выходного диапазона ЦАП. Передаваемые параметры: unsigned char id, unsigned char dac – номер канала ЦАП (0 – A_{OUT} 0, 1 – A_{OUT} 1), diap - код диапазона. Значения кодов представлены в таблице справа. Соответствие кодов и выходных напряжений для различных типов каналов представлено в таблице ниже (U_{пр} – предельное напряжение канала).

| Код | Однополярный режим | Биполярный режим |
|-------|--------------------|------------------|
| -2048 | 0 | -U _{пр} |
| 2047 | U _{пр} | U _{пр} |

| Значение diap | Выходной диапазон канала, В |
|---------------|-----------------------------|
| 0 | 0..5 |
| 1 | 0..10 |
| 2 | 0..10.8 |
| 3 | -5..5 |
| 4 | -10..10 |
| 5 | -10.8..10.8 |

bool __stdcall SetDacValue(unsigned char id, unsigned char dac, int val)

Однократный вывод данных на ЦАП. Преобразование выполняется в момент обращения к данной функции из приложения. Передаваемые параметры: unsigned char id, unsigned char dac – установка номера канала (0 – A_{OUT} 0, 1 – A_{OUT} 1), int val – переменная, содержащая преобразуемый код.

bool __stdcall PowerUpDac(unsigned char id, unsigned char dac)

Включение канала ЦАП после срабатывания защиты от перегрузки по току. Передаваемые параметры: unsigned char id, unsigned char DAC – номер ЦАП (0 – A_{OUT} 0, 1 – A_{OUT} 1).

bool __stdcall ClearDACs(unsigned char id)

Сброс каналов ЦАП (установка на обоих каналах нулевого напряжения). Передаваемый параметр – unsigned char id.

Функции работы со счетчиком событий

bool __stdcall EnableCounter(unsigned char id, BOOL enable)

Запуск либо останов счетчика событий. Передаваемые параметры: unsigned char id, BOOL Enable (TRUE – запуск, FALSE – останов).

bool __stdcall ReadCounter(unsigned char id, unsigned int *data)

Чтение значений счетчика событий. Передаваемые параметры: unsigned char id, и указатель на переменную, принимающую данные unsigned int *data.

bool __stdcall ClearCounter(unsigned char id)

Сброс счетчика событий. Передаваемый параметр – unsigned char id.

Функции дискретного В/В

bool __stdcall ConfigureDigitalPort(unsigned char id, unsigned char data)

Конфигурация линий дискретного В/В D_{IO} A0..A7 для использования в однократном режиме. Передаваемые параметры: unsigned char id, unsigned char data (состояние бита для каждой линии: 0 – вывод, 1 – ввод).

bool __stdcall ReadDIValue(unsigned char id, unsigned char *data)

Однократный ввод данных из порта дискретного В/В. Передаваемые параметры: unsigned char id, и указатель на переменную, принимающую байт данных unsigned char *data.

bool __stdcall SetDOValue(unsigned char id, unsigned char data)

Однократный вывод данных в порт дискретного В/В. Передаваемые параметры: unsigned char id и данные unsigned char data.

bool __stdcall SetFastDIOFreq(unsigned char id, unsigned long freq, unsigned char DOorDI)

Установка частоты В/В дискретных сигналов. Значение частоты задается с помощью внутреннего таймера в диапазоне 50 кГц..8 МГц (верхняя граница диапазона в действительности может быть меньше из-за низкой производительности PC). Передаваемые параметры: unsigned char id, unsigned long freq – частота В/В (50000..8000000), unsigned char DOorDI – переменная для задания частоты цифрового ввода (DOorDI = 1) либо вывода (DOorDI = 2).

bool __stdcall FastReadDI(unsigned char id, unsigned char *InBuf, LONG Length, unsigned long Timeout, bool WaitComplete = true)

Ввод массива данных из порта дискретного В/В D_{IO} A0..A7. Передаваемые параметры: unsigned char id, unsigned char *InBuf – указатель на массив, принимающий данные; LONG Length – размер массива (должен быть кратным 512 и не должен превышать $2 \cdot 10^6$), unsigned long Timeout – таймаут ожидания окончания чтения данных (в ms, по умолчанию равен 1000, при превышении установленного значения чтение АЦП прерывается), bool WaitComplete – режим ожидания данных (по умолчанию TRUE). При WaitComplete = TRUE функция ожидает завершения передачи данных, в противном случае выход из функции происходит сразу после начала передачи и пользователь имеет возможность обрабатывать ранее полученные данные (по умолчанию TRUE). Для проверки окончания передачи данных (при WaitComplete = FALSE) необходимо использовать функцию isReadDIXferComplete.



Во время передачи данных с дискретных входов обращение к устройству (кроме isReadDIXferComplete) не допустимо.



Для корректной работы функции необходимо равенство переданного с помощью Length значения, и размера массива InBuf.

```
bool __stdcall FastWriteDO(unsigned char id, unsigned char *OutBuf, LONG Length, unsigned long Timeout, bool WaitComplete = true)
```

Вывод массива данных в порт дискретного В/В D_{IO} A0..A7. Передаваемые параметры: unsigned char id, unsigned char *pOutBuf – указатель на массив данных, заданных пользователем; LONG Length – размер массива (должен быть кратным 512 и не должен превышать $2 \cdot 10^6$), unsigned long Timeout – таймаут ожидания окончания чтения данных (в ms, по умолчанию равен 1000, при превышении установленного значения вывод данных прерывается), bool WaitComplete – режим ожидания данных (по умолчанию TRUE). При WaitComplete = TRUE функция ожидает завершения передачи данных, в противном случае выход из функции происходит сразу после начала передачи и пользователь имеет возможность обрабатывать ранее полученные данные (по умолчанию TRUE). Для проверки окончания передачи данных (при WaitComplete = FALSE) необходимо использовать функцию isWriteDOXferComplete.



Во время передачи данных на дискретные выходы обращение к устройству (кроме isWriteDOXferComplete) не допустимо.



Для корректной работы функции необходимо равенство переданного с помощью Length значения и размера массива OutBuf.

```
bool __stdcall isReadDIXferComplete(unsigned char id, unsigned char *InBuf)
```


Проверка, закончена ли передача данных от устройства к PC при выполнении функции FastReadDI. Передаваемые параметры: unsigned char id, unsigned char *InBuf – указатель на массив данных, используемый в функции FastReadDI. Возвращаемое значение: FALSE – передача данных не закончена, TRUE – передача данных закончена.

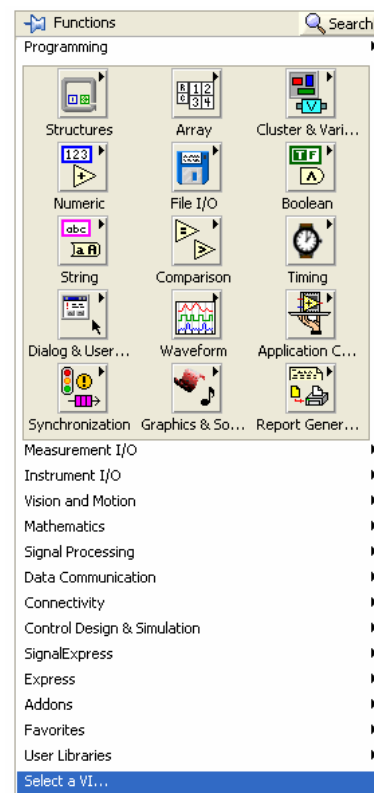
```
bool __stdcall isWriteDOXferComplete(unsigned char id)
```

Проверка, закончена ли передача данных от PC к устройству при выполнении функции FastWriteDO. Передаваемый параметр – unsigned char id. Возвращаемое значение: FALSE – передача данных не закончена, TRUE – передача данных закончена.

Библиотека подпрограмм m-DAQ12 library.llb

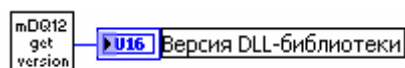
Для использования подпрограмм библиотеки в LabVIEW-программе необходимо выбрать пункт **Select a VI...** палитры функций панели блок-диаграммы, после чего указать местонахождение **m-DAQ12 library.llb** и требуемую подпрограмму.

 В случае получения следующей ошибки ОС при работе с LabVIEW-библиотекой: «Приложение не было запущено, поскольку оно некорректно настроено. Повторная установка приложения может решить данную проблему», необходимо предварительно установить распространяемый пакет Microsoft Visual C++ 2008 SP1 Redistributable Package из папки C++ 2008 SP1\ поставляемого диска.



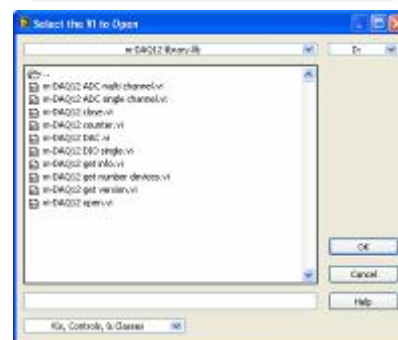
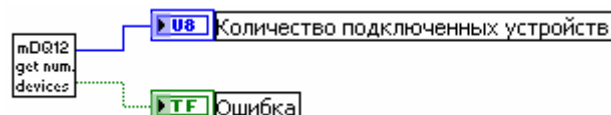
m-DAQ12 get version.vi

Получение номера версии DLL-библиотеки.



m-DAQ12 get number devices.vi

Получение количества подключенных к PC устройств.



Здесь и далее значение, возвращаемое на выходе **Ошибка** (TRUE либо FALSE) предназначено для контроля успешного выполнения subVI.

m-DAQ12 open.vi

Открытие подключенного к шине USB устройства.



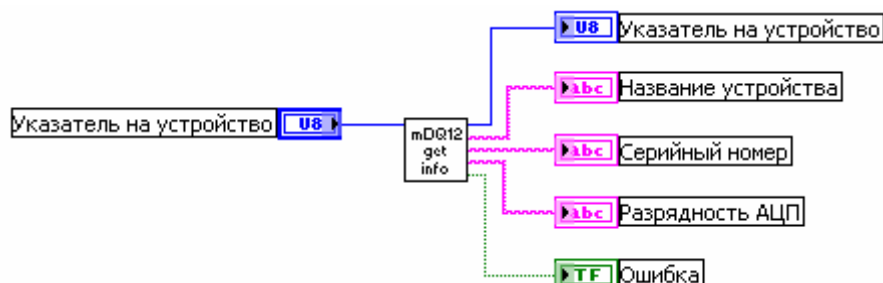
Вход **Номер устройства** задает номер модуля, который необходимо открыть (нумерация начинается с 0 в порядке подключения устройств к USB). Здесь и далее **Указатель на устройство** – переменная, с помощью которой производится последующая адресация команд.

m-DAQ12 close.vi

Закрытие подключенного к шине USB устройства.

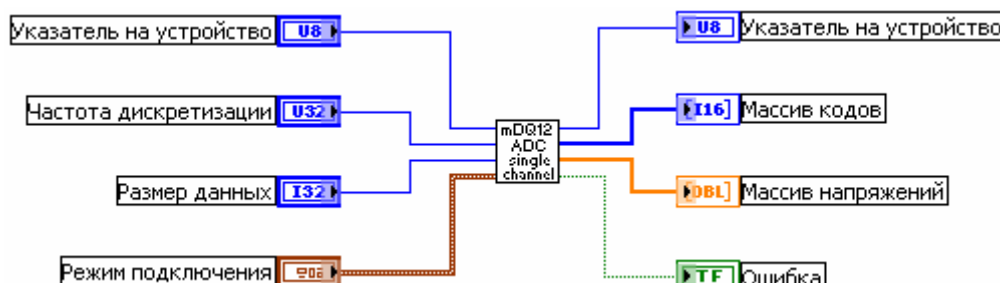
**m-DAQ12 get info.vi**

Получение информации о подключенном к шине USB устройстве.



m-DAQ12 ADC single channel.vi

Циклический опрос одного из входов АЦП.



На вход **Частота дискретизации** необходимо подать значение соответствующей частоты, Гц, по умолчанию оно равно 100000 Гц. На вход **Размер данных** необходимо подать значение соответствующей величины (кратное 512, по умолчанию – 512). На вход **Режим подключения** необходимо подать кластер, содержащий три целочисленных переменных в формате **U8** (см. соответствующую таблицу ниже). Выходы **Массив кодов** и **Массив напряжений** (в В) содержат полученные данные.

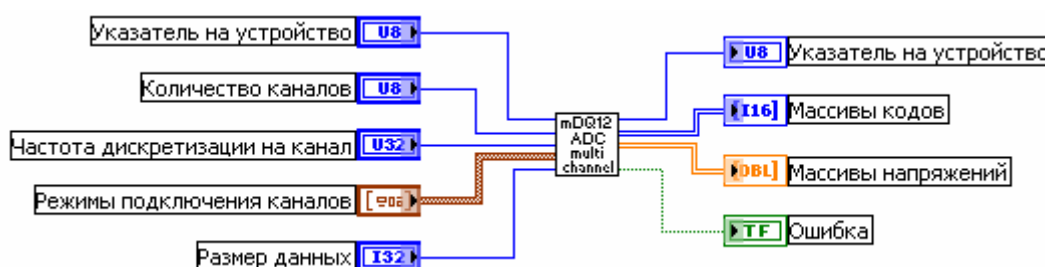
| Содержание кластера <i>Режим подключения</i> | | |
|--|----------------------------|--|
| Переменная | Описание | Диапазон устанавливаемых значений |
| Номер | Номер опрашиваемого канала | для дифф. подключения – 0..3; для однопроводного подключения – 0..7 |
| Режим | Режим подключения | 0 – режим с заземленным входом; 1 – дифф. подключение; 2 – однопроводное подключение |
| Диапазон | Входной диапазон АЦП | 1 – 0..10 В; 2 – 0..5 В; 3 – 0..2,5 В; 4 – 0..1,25 В |



Подпрограммы **m-DAQ12 ADC single channel.vi** и **m-DAQ12 ADC multi channel.vi** непригодны для чтения данных в режиме реального времени.

m-DAQ12 ADC multi channel.vi

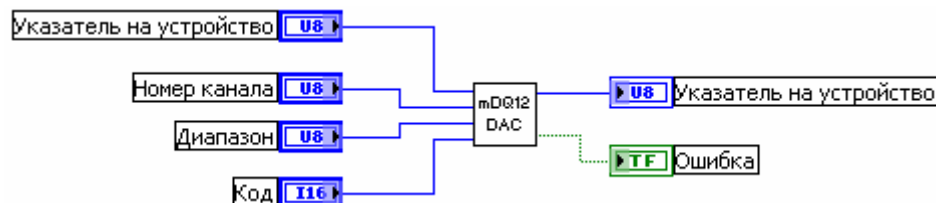
Циклическое считывание данных с нескольких входов АЦП.



От subVI **m-DAQ12 ADC single channel.vi** отличается следующими входами: частота дискретизации устанавливается из расчета на один канал с использованием входа **Частота дискретизации на канал**; на вход **Режимы подключения каналов** необходимо подать массив кластеров, идентичных переменной **Режим подключения**. Максимальное количество устанавливаемых каналов – 8, очередность их опроса соответствует очередности задания кластеров в массиве. Выход **Массивы кодов** содержит матрицу полученных данных, каждая строка которой – массив кодов с отдельного входа АЦП. Выход **Массивы напряжений** содержит аналогичную матрицу полученных данных, в В.

m-DAQ12 DAC.vi

Вывод данных на ЦАП.

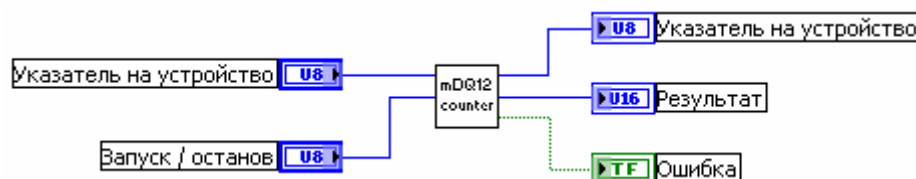


Вход **Номер канала** определяет канал ЦАП: A_{OUT} 0 (0, по умолчанию), A_{OUT} 1 (1). На вход **Диапазон** нужно подать значение для программной установки выходного диапазона канала ЦАП (см. соответствующую таблицу справа). На вход **Код** необходимо подать соответствующее значение в диапазоне -2048..2047, по умолчанию – 0.

| Значение переменной Диапазон | Выходной диапазон ЦАП |
|---------------------------------|--------------------------|
| 0 | 0..5 В |
| 1 | 0..10 В |
| 2 | 0..10,8 В |
| 3 | -5..5 В |
| 4 | -10..10 В |
| 5 | -10,8..10,8 В |

m-DAQ12 counter.vi

Счетчик событий.



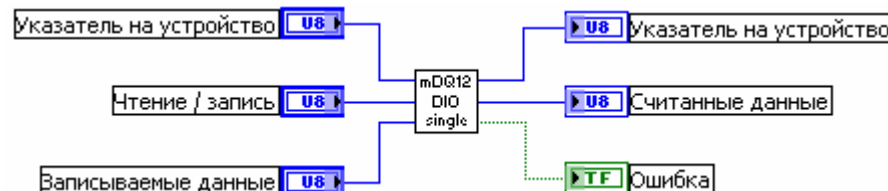
Счетчик событий срабатывает по переднему или заднему фронту входного сигнала в зависимости от состояния входа **Фронт** (передний – 0, задний – 1). Его значение по умолчанию – 0.

Вход **Запуск / останов** задает временной интервал счета от момента установки на нем 0 (запуск) до момента установки 1 (останов). Корректный результат счета может быть получен только после останова счетчика.

Выход **Результат** содержит результат счета.

m-DAQ12 DIO single.vi

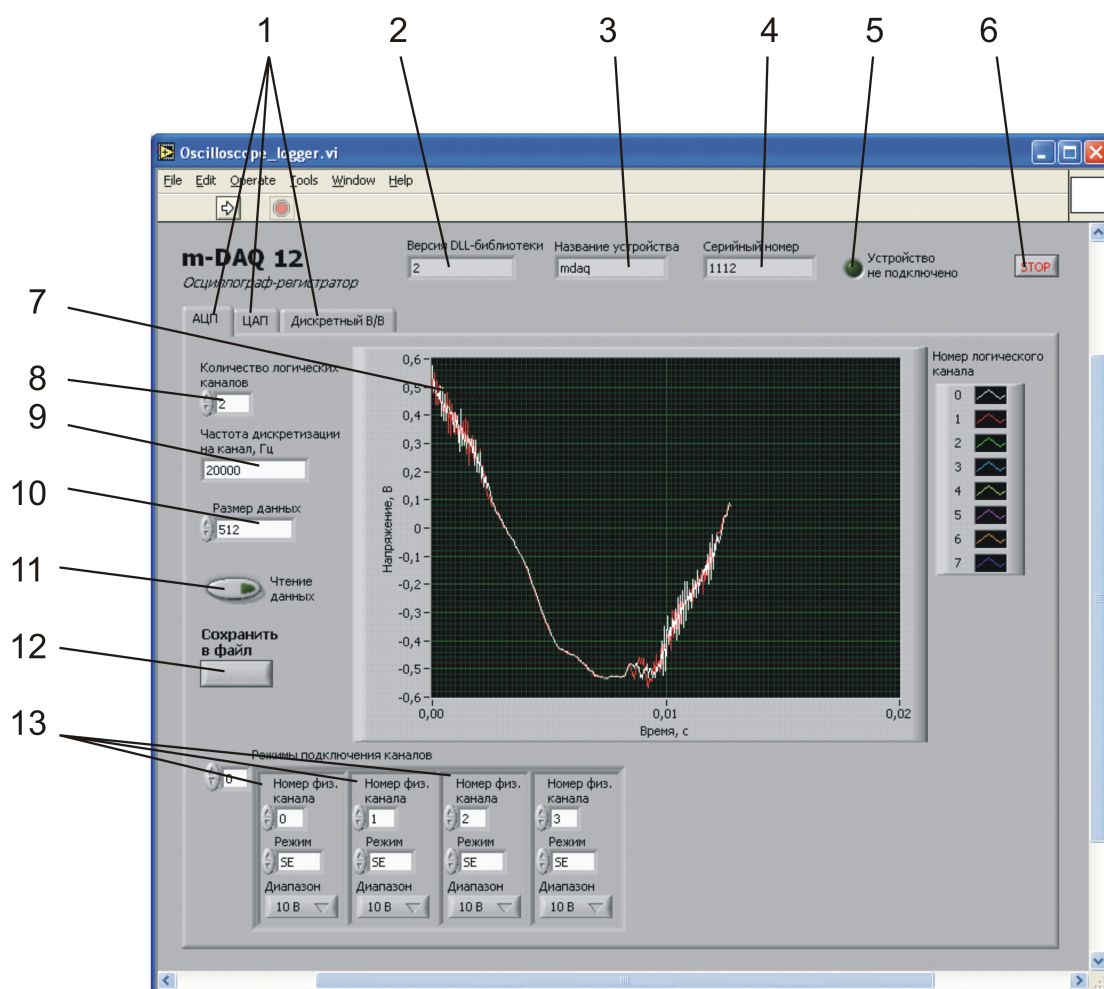
Чтение / запись состояний линий портов дискретного В/В.

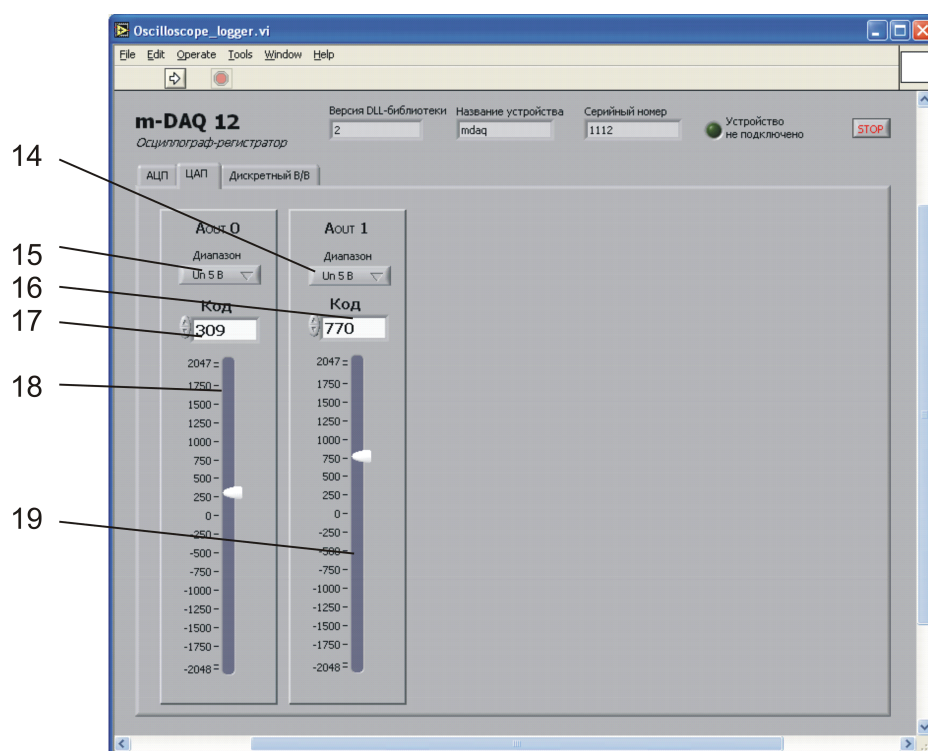


Вход **Чтение / запись** определяет направление данными: чтение – 0 (по умолчанию), запись – 1.

Oscilloscope_logger.exe

Для функционирования виртуального осциллографа на основе m-DAQ12/m-DAQ14 необходимы LabVIEW 8.2.1 или LabVIEW Run-Time Engine 8.2.1. Виртуальный осциллограф рассчитан на подключение к компьютеру одного m-DAQ12/m-DAQ14.





Последовательность работы с виртуальным осциллографом-регистратором**Подготовка к работе**

1. Включить компьютер.
2. Присоединить с помощью поставляемого кабеля связи по USB модуль *m-DAQ12* либо *m-DAQ14* к компьютеру.
3. Используя кабельную часть разъема DB 25-M, подключить источники либо приемники сигналов к соответствующим каналам модуля.
4. Загрузить с использованием приложения *LabVIEW 8.2.1* либо *LabVIEW Run-Time Engine 8.2.1* исполняемый файл *Oscilloscope_logger.exe*. Программа будет автоматически запущена на выполнение в циклическом режиме.

В случае, если устройство не было опознано программной средой (при свечении соответствующего индикатора 5) необходимо проверить корректность его подключения либо установки драйвера. После чего запустить программу на выполнение.

5. Остановка программы осуществляется с помощью кнопки 6.

Управление АЦП

1. Перейти на закладку «АЦП» программы, нажав на ее название в перечне закладок 1.
2. Ввести в поле 8 количество N_{ch} опрашиваемых каналов (диапазон возможных значений N_{ch} - 1..8), а в поле 10 – размер считываемой за один цикл работы выборки данных (задается в отсчетах).
3. Ввести в поле 9 частоты опроса требуемое значение F_{s_1ch} частоты дискретизации входного сигнала (из расчета на один канал). Данную процедуру необходимо выполнять до старта чтения данных. При необходимости изменения частоты дискретизации необходимо остановить чтение. В случае $N_{ch} * F_{s_1ch} > 100$ кГц выполняется автоматическая установка $F_{s_1ch} = 100$ кГц / N_{ch} .
4. С помощью полей ввода 13 установить требуемые параметры чтения для опрашиваемых каналов (номер входа АЦП, режим подключения, входной диапазон).
5. Перевести АЦП в режим циклического чтения данных путем нажатия кнопки 11. Визуальное представление данных осуществляется с помощью осциллографа 7. Останов АЦП осуществляется с помощью кнопки 6.
6. При необходимости записи результатов работы АЦП в файл нажать соответствующую кнопку 12. После нажатия кнопки указать путь и название файла (расширение по умолчанию - *.dat, возможно задание расширения пользователем).

Результаты записываются в текстовый файл в следующем виде. Данные с каждого логического канала содержатся в отдельном столбце (количество столбцов в файле соответствует количеству выбранных каналов). Первые 4 значения в столбце содержат параметры чтения (описание параметров представлено в соотв. таблице ниже), 5-е и последующие значения – массив считанных данных (полученные коды).

Содержание столбца файле результатов чтения с АЦП

| Положение в столбце | Описание | Диапазон устанавливаемых значений |
|---------------------|-------------------------|--|
| 1 | Номер входа | для дифф. подключения – 0..3; для однопроводного подключения – 0..7 |
| 2 | F_{s_1ch} | 0..100000 |
| 3 | Режим подключения | 0 – режим с заземленным входом; 1 – дифф. подключение; 2 – однопроводное подключение |
| 4 | Входной диапазон | 1 – 0..10 В; 2 – 0..5 В; 3 – 0..2,5 В; 4 – 0..1,25 В |
| 5 и ниже | Считанные данные (коды) | - |

- После записи результатов в файл необходим повторный запуск чтения данных.
- В процессе работы возможна смена всех установленных ранее настроек, для чего необходим останов АЦП.

Управление ЦАП

- Перейти на закладку «ЦАП» программы, нажав на ее название в перечне закладок 1.
- Установить выходные диапазоны каналов ЦАП $A_{out\ 0}$ и $A_{out\ 1}$ с помощью выпадающих списков 15 и 14 соответственно.
- Установить преобразуемые коды с помощью элементов управления 16 – 19 (ползунки и соответствующие поля цифрового ввода синхронизированы).

Управление линиями дискретного В/В

- Перейти на закладку «Дискретный В/В» программы, нажав на ее название в перечне закладок 1.
- Выбрать направление передачи данных с помощью переключателя 20.
- Для записи данных на дискретные линии использовать кнопки 21 (записываемые данные представляются в десятичном цифровом виде в поле 23).
- Считанные данные представляются с помощью световых индикаторов 22 и в десятичном цифровом виде в поле 24.