

LxDCS - ПАКЕТ ДЛЯ РАЗРАБОТКИ РАСПРЕДЕЛЕННЫХ СИСТЕМ УПРАВЛЕНИЯ

2. БАЗОВЫЕ ЭЛЕМЕНТЫ СИСТЕМЫ LxDCS

2.1. Регистры данных

Основным элементом хоста являются регистры данных, предназначенных для хранения текущих значений технологических данных, уставок, временных и рабочих переменных и управляющих сигналов. Регистры доступны для чтения и записи для различных приложений LxDCS. Набор регистров представляет собой три массива, размещенных в оперативной памяти LxDCS-хоста:

- Аналоговые регистры (**ARn**), каждый регистр занимает 4 байта, тип данных - float. Предназначены для хранения значений аналоговых параметров. Максимально возможное количество для одного хоста - 65535.
- Дискретные регистры (**DRn**), каждый занимает один байт, формат данных - unsigned char. Предназначены для хранения значений дискретных параметров, а так же небольших целых величин. Максимально возможное количество для одного хоста - 65535.
- Служебные регистры (**SRn**, Status Register), каждый занимает один байт, формат - unsigned char. Предназначены для хранения системной информации - текущего времени и флагов управления приложениями. Количество регистров - 16..

2.2. Общая конфигурация

Общие параметры системы описываются главным файлом конфигурации **/etc/lxdcs.conf**, Этот файл используют все приложения lxdcs.

Директивы файла конфигурации **lxdcs.conf**

analog=num

Задает число аналоговых регистров **AR**. Если директива отсутствует в файле конфигурации, число регистров устанавливается равным 16.

discret=num

Задает число дискретных регистров **DR**. Если директива отсутствует в файле конфигурации, число регистров устанавливается равным 16.

shmkey=filename

Задает ключ доступа к регистрам. Ключом является имя любого существующего в системе файла с указанием полного пути. Если директива отсутствует, в качестве ключа используется сам файл **/etc/lxdcs.conf**

datapath=path

Задается полный путь к корневой директории LxDCS, где размещаются файлы конфигурации приложений и база данных. Конфигурации приложений помещаются в директорию **/datapath/etc**, архивы данных - в **/datapath/data**.

3. СИСТЕМНЫЕ УТИЛИТЫ

Для непосредственной работы с регистрами предназначены следующие приложения:

- Консольная утилита чтения/записи регистров.
- Симулятор данных, предназначен для динамического генерирования значений регистров данных.
- Вычислитель, предназначен для обработки данных, размещенных в регистрах. Может выполнять арифметические и логические операции, а так же некоторые специальные функции.
- Утилита управления работой приложений.

3.1. Утилита чтения-записи регистров данных **lc_rrw**

Консольная утилита **lc_rrw** предназначена для просмотра и записи регистров аналоговых и дискретных данных а так же служебной информации.

Командная строка:

```
lc_rrw [-v] [-r] [-t text] [An[:num][%format]].. [An=val]..  
      [Dn[:num][%format]].. [Dn=val] [Sn[:num][%format]].. [Sn=val]..
```

Опции командной строки:

- | | |
|---------|---|
| -v | - при чтении выводит тип и номер регистра (e.g. A12=1.2345) |
| -n | - вставляет в выходной поток символ перевода строки |
| -t text | - вставляет в выходной поток text. text может быть заключен в кавычки. |
| An | - чтение регистра аналоговых данных с номером n |
| Dn | - чтение регистра дискретных данных с номером n |
| Sn | - чтение регистра статуса с номером n |
| :num | - множественное чтение num регистров |
| %format | - формат вывода значений регистров, аналогичный принятому в языке Си (e.g. %5.2f, %03d) |
| An=val | - запись в регистр аналоговых данных n значения val (float) |
| Dn=val | - запись в регистр дискретных данных n значения val (0..255) |
| Sn=val | - запись в служебный регистр n значения val (0..255) |

Приложение может использоваться в составе shell скриптов для простых форм отображения текущих данных или для установки начальных значений регистров при старте системы.

3.2. Симулятор данных - lc_simd

Демон xsio_simd предназначен для периодического генерирования статических или изменяющихся значений регистров аналоговых (AR) и дискретных (DR) данных. Режимы симуляции задаются файлом конфигурации, находящимся в директории /datapath/etc/. Сообщения записываются в файл журнала /var/log/lc_simd.cfgfile.log, где cfgfile - имя файла конфигурации.

Командная строка:

```
lc_simd [-c cfgfile] [--debug]
lc_simd --help
```

Опции командной строки:

- c cfgfile - задает файл конфигурации file, если не указано, используется файл xsio_simd.conf
- debug - отладочный режим. На стандартный вывод отображаются результаты генерации данных.
- help - вывод подсказки

Директивы файла конфигурации:

period=time

Определяет периодичность симуляции данных в миллисекундах. Если опция не определена, устанавливается в 1000. Если time=0, приложение завершается после выполнения одного цикла генерации. time=0 может использоваться для начальных уставок при старте системы.

cfgreload=time

определяет периодичность перезагрузки файла конфигурации в минутах. Если опция не определена, перезагрузка производится каждые 5 минут. Если n=0, перезагрузка не производится.

An=value

В аналоговый регистр n записываются статические данные value (float).

An=random data=min:max step=delta delay=tick

В аналоговый регистр записываются случайные данные в диапазоне от min до max (float), с шагом изменения delta (float). Значение tick (int) определяет период изменения данных, который будет равен значению period*tick

Dn=value

В дискретный регистр n записываются статические данные value (int 0..255).

Dn=random data=min:max step=delta delay=time

В дискретный регистр записываются случайные данные в диапазоне от min до max (float), с шагом изменения delta (float). Значение tick (int) определяет период изменения данных, который будет равен значению period*tick

Пример файла конфигурации:

```
# lc_simd.conf - data simulator configuration
#
period=100
cfgreload=60
#
A1=random data=24.3:100.23 step=12.34 delay=100
A0=923.456
A2=-234.567
D123=43
A5=+00.3
D18=random data=2:35 step=3 delay=100
A7=0.009
A9=random data=-25.7:100.70 step=0.15 delay=10
D79=1
```

3.3. Управление приложениями - lc_stop

Утилита предназначена для завершения работы всех процессов подсистем LxDCS и удаления сегментов разделяемой памяти.

Командная строка

```
lc_stop [options]
lc_stop [--help]
```

Опции командной строки:

- io - останов подсистемы ввода-вывода
- arch - останов архивирования данных
- net - останов сетевой подсистемы

Запуск утилиты без опций приводит к завершению всех процессов LxDCS и освобождению сегментов разделяемой памяти, занятых регистрами данных.