

LxDCS - ПАКЕТ ДЛЯ РАЗРАБОТКИ РАСПРЕДЕЛЕННЫХ СИСТЕМ УПРАВЛЕНИЯ

14 ноября 2013 г.

4. ПОДСИСТЕМА ВВОДА _ ВЫВОДА

4.1. Назначение и состав

Подсистема ввода-вывода предназначена обмена данными с различными устройствами связи с объектом, их обработки и архивирования полученных данных. Подсистема включает следующие программы:

- - Драйвер MODBUS RTU, предназначен для обмена данными между регистрами и устройствами с протоколом MODBUS RTU по последовательному интерфейсу.
- - Консольная утилита для работы с устройствами MODBUS по последовательному интерфейсу
- - Драйвер модулей ввода-вывода ICP I-7000, предназначен для обмена данными между регистрами и устройствами с протоколом DCON по последовательному интерфейсу.

4.2. Драйвер MODBUS RTU lcio_modbusd.

Демон lcio_modbusd предназначен для работы с различными устройствами, имеющими интерфейс MODBUS RTU в качестве мастер-устройства (клиент). Программа производит циклическое чтение и запись данных в подчиненное устройство, подключенное по последовательному интерфейсу. Результаты чтения регистров и логических ячеек записываются соответственно в регистры данных **Rn**, оттуда же берутся данные для вывода. К данным может применяться масштабирование. Программа также может выполнять некоторые операции над данными регистров. Работа программы описывается в файле конфигурации, находящимся в директории `/$datapath/etc/`. Сообщения записываются в файл журнала `/var/log/lcio_modbusd.cfg.log`, где `cfg` - имя файла конфигурации.

Командная строка :

```
lcio_modbuscd -c cfgfile [--debug]
```

Опции командной строки:

- **-c cfgfile** - задает файл конфигурации
- **--debug** - выдача результатов работы программы на консоль.

Директивы файла конфигурации:

- **delay=time** - Определяет задержку цикла выполнения в миллисекундах. Если директива отсутствует, устанавливается в 1000.
- **cfgreload=time** Определяет периодичность перезагрузки файла конфигурации в минутах. Если опция не определена, перезагрузка производится каждые 5 минут. Если директива отсутствует или значение равно нулю, перезагрузка не производится.
- **iodev=dev iopar=br:cs:pb:sb** - Определяет параметры интерфейса. **dev** - последовательный порт (e.g. /dev/ttyS0), **br** - скорость (1200..115200), **cs** - число бит (7|8), **pb** - бит паритета (E|N|E), **sb** - стоповый бит (1|2)
- **in_mhr=addr:off:num:Rerr** - Чтение с устройства с адресом **addr** **num** регистров хранения (FC3) начиная с **off**, в регистр **Rerr** записывается результат выполнения операции ("0"-успешное, "1"-ошибка). В случае успешного считывания, принятые значения MODBUS-регистров размещаются в буфере приема и именуются **M0..Mn**. Следом за директивой идет блок, описывающий обработку принятых данных:
- **Rn=Mn input=min:max scale=smin:smax** - данные из буфера **Mn**, записываются в регистр данных **LxDCS Rn**, предварительно пройдя преобразование значения из диапазона считанных данных **min** и **max** (MODBUS 0..65535,) диапазон локального регистра **Rn**, значение которого определяется **smin** и **smax** (float). В случае ошибки чтения данные регистров не изменяются.
- **end** - завершает блок описания обработки принятых данных.

Пример:

```
in_mhr=0x2F:40027:3:R7
R8=M0 input=200:16000 scale=50:200
R12=M1 input=0:65535 scale=-6.3:12
R17=M2 input=1000:4500 scale=0:600
end
```

Чтение с устройства, имеющего адрес 0x2F трех регистров хранения MODBUS начиная с 40027, первый считанный регистр помещается в регистр R8 с предварительным приведением из диапазона входных значений 200..16000, в диапазон от 50 до 200. Результат выполнения записывается в регистр R7.

- **out_mhr=addr:off:num:Rerr** - Запись в устройство с адресом **addr** **num** регистров хранения (FC3) начиная с **off**. В регистр **Rerr** записывается результат выполнения операции ("0"-успешное, "1"-ошибка). Данные для записи размещаются в буфере передачи и именуются **M0..Mn**. Следом за директивой идет блок, описывающий обработку данных перед передачей.
- **Mn=Rn output=min:max scale=smin:smax** - Данные в буфере передачи **Mn** берутся из регистра **LxDCS Rn**, предварительно пройдя преобразование из диапазона лрегистра **Rn**, значение которого определяется **smin** и **smax** (float) в диапазон выходных значений **min** и **max** (MODBUS 0..65535,).
- **end** - завершает блок обработки отправляемых данных

Пример:

```
out_mhr=0x2F:40027:3:R007
M0=R17 output=200:16000 scale=50:200
M1=R23 output=0:65535 scale=-6.3:12
M2=R09 output=1000:4500 scale=0:600
end
```

Запись в устройство, имеющего адрес 0x2F трех регистров хранения начиная с 40027, в который записывается значение регистра R17, приведенное в диапазон 200 до 16000 из 50..200. следующие данные регистров R23 и A9 записываются в регистры 40028 и 40029 соответственно. Результат выполнения записывается в дискретный регистр R7.

Арифметические операции, позволяют производить элементарную обработку данных

- **Rn=val** - Операция присваивания константы - регистру **Rn** присваивается значение **val**
- **Rn=RN** - Операция копирования регистру **Rn** присваивается значение регистра **RN**
- **Rn+=RN** - Операция сложения - к значению аналогового регистра **Rn** прибавляется значение **RN**, результат записывается в регистр **Rn**
- **Rn-=RN** - Операция вычитания - из значения аналогового регистра **Rn** вычитается значение **RN**, результат записывается в регистр **Rn**
- **Rn*=RN** - Операция умножения - значение регистра **Rn** умножается на значение **RN**, результат записывается в регистр **Rn**
- **Rn=/RN** - Операция деления - значение регистра **Rn** делится на значение **RN**, результат записывается в регистр **Rn**

- **Rn=sqrt(RN)** - Операция извлечения квадратного корня - вычисляется квадратный корень значение регистра **RN**, результат записывается в **Rn**

4.3. Консольный клиент MODBUS RTU **lcio_mbc**

Консольная утилита **lcio_mbc** предназначена для чтения и записи регистров и логических ячеек устройств, имеющих протокол MODBUS RTU, подключенных через последовательный интерфейс.

Командная строка:

```
lcio_mbc sp:br:addr [-v] [-r] [-t text] [Rn[:num][%format]].. [Rn=val]..
[Dn[:num][%format]].. [Dn=val] ..
```

Опции командной строки:

- **sp** - устройство последовательного порта (e.g. ttyS0)
- **br** - скорость передачи данных (e.g. 9600)
- **addr** - MODBUS адрес устройства
- **-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.
- **Rn** - чтение регистра с номером n
- **Dn** - чтение логической ячейки с номером n
- **:num** - множественное чтение num регистров или логических ячеек
- **%format** - формат вывода значений регистров, аналогичный принятому в языке Си, допускаются форматы %[d, %[x, %[X, %b, последний выводит содержимое регистра в бинарном виде, начиная со старшего бита
- **Rn=val** - запись в регистр n значения val (int).
- **Dn=val** - запись в логическую ячейку n значения val (0/1)

4.4 Драйвер модулей ICP I70XX - **lcio_i7kd**

Демон **lcio_i7kd** предназначен для работы с модулями аналогового и дискретного ввода-вывода ICP DAS серии I7000 и аналогичными, имеющими систему команд DCON (ADAM4000, NuDam6000...). Программа производит циклическое чтение и запись данных. Результаты ввода записываются в регистры данных, оттуда же берутся данные для вывода. К данным может применяться масштабирование. Программа также может выполнять некоторые операции над данными регистров. Работа программы описывается в файле конфигурации, находящимся в директории `/datapath/etc/`. Сообщения записываются в файл журнала `/var/log/lcio_i7kd.cfg.log`, где `cfg` - имя файла конфигурации.

Командная строка

```
lcio_i7kd -c cfgfile [--debug]
```

Опции командной строки:

- **-c cfgfile** - задает файл конфигурации
- **--debug** - выдача результатов работы программы на консоль.

Опции файла конфигурации:

- **delay=time** - Определяет задержку цикла выполнения в миллисекундах. Если директива отсутствует, устанавливается в 1000.
- **cfgreload=time** Определяет периодичность перезагрузки файла конфигурации в минутах. Если опция не определена, перезагрузка производится каждые 5 минут. Если директива отсутствует или значение равно нулю, перезагрузка не производится.
- **iodev=dev iopar=br:cs:pb:sb** - Определяет параметры интерфейса. **dev** - последовательный порт (e.g. `/dev/ttyS0`), **br** - скорость (1200..115200), **cs** - число бит (7|8), **pb** - бит паритета (E|N|E), **sb** - стоповый бит (1|2)
- **in_17s:addr:ch:Rerr data=Rdata input=min:max scale=smin:smax** - Чтение данных с модуля аналогового ввода I7017 с адресом **addr**, канал **ch**, данные записываются в регистр **Rdata**. Диапазон входного сигнала **max** и **min** в единицах модуля, диапазон шкалы данных **smin** и **smax**. В регистр **Rerr** записывается результат выполнения операции ("0"-успешное, "1"-ошибка). В случае ошибки содержимое регистра **Rdata** не изменяются. Данная команда так же может использоваться для чтения данных с модулей 7018,7033.

Пример:

```
in_17s:0x2F/4:R07 data=R08 input=4:20 scale=50:200
```

Чтение в регистр R8 данных модуля с адресом 0x2F, канал 4, диапазон входного сигнала 4..20, шкала от -50 до 200, результат выполнения записывается в дискретный регистр R7

- **in_60m:addr:Rerr data=Rd0:Rd1:Dd2:Dd3** - Чтение 4х входов модуля дискретного ввода-вывода I7060 с адресом **addr**, считанные данные записываются в регистры Rd0..Rd3. В регистр **Rerr** записывается результат выполнения операции ("0"-успешное, "1"-ошибка). Данная команда так же может использоваться с модулями

Пример:

in_60m:0x21:R29 data=R13:R14:R15:R16

Чтение данных модуля с адресом 0x21, данные записываются в регистры R13,R14,R15,R16, результат выполнения записывается в дискретный регистр D29

- **in_60c:addr/ch:Rerr data=Rd scale=k** - Чтение счетчика дискретного входа модуля I7060 с адресом **addr**, дискретного входа **ch**, данные, умноженные на коэффициент K, записываются в регистр Rd. В регистр **Rerr** записывается результат выполнения операции ("1"-успешное, "0"-ошибка).

Пример:

in60c/0x54:2:R17 data=R24 scale=31.5

Чтение данных счетчика модуля с адресом 0x54, вход 2, данные, умноженные на 31.5, записываются в регистр R24, результат выполнения записывается в регистр R17

- **out60m:addr:Rerr data=Rd0:Rd1:Rd2:Rd3** - Вывод данных регистров Dn0..Dn3 на модуль дискретного ввода-вывода I7060 с адресом **addr**. В регистр **Rerr** записывается результат выполнения операции ("1"-успешное, "0"-ошибка)

Пример:

out60m:0x31:R12 data=R43:R8:R0:R123

Вывод данных регистров R43, R8, R0, R8

- **out44m:addr:Rerr data=Rn0:Rn1:Rn2:Rn3:Rn4:Rn5:Rn6:Rn7** - Вывод данных регистров Rn0..Rn7 на модуль дискретного ввода-вывода

I7044 с адресом **addr**. В регистр **Rerr** записывается результат выполнения операции ("1"-успешное, "0"-ошибка)

Пример:

`out44m:0x31:R12 data=R43:R8:R0:R123:R67:R12:R2080:R4`

Вывод данных дискретных регистров R43:R8:R0:R123:R67:R12:R2080:R4

- `out24s:addr:ch,Rerr data=Rdata output=min:max scale=smin:smax`
- Аналоговый вывод с модуля I702X с адресом **addr**, канал **ch**, данные для вывода считываются из регистра **Rdata**. Диапазон данных **max** и **min** в единицах физической величины, диапазон выходных значений тока (напряжения) **smin** и **smax**. В дискретный регистр **Dn** записывается результат выполнения операции ("1"-успешное, "0"-ошибка). Данная команда может использоваться для модулей 7021, 7023

Пример:

`out24s:0x2F:2:R8 data=R12 output=0:20 scale=-100:1000`

Установка значения выходного тока модуля с адресом 0x2F, канал 2, диапазон физической величины, находящейся в регистре R12 от -100 до 1000, диапазон выходного тока 0-20mA, результат записывается в регистр R8

Программа также может выполнять простую обработку данных. Директивы арифметических операций соответствуют приведенным в п 4.2