

CPBK DevLink

**Драйвер протокола
МЭК 870-5-2-95 (FT3) КП**

Версия 1.2

Руководство Пользователя

CPBK DevLink. Драйвер протокола МЭК 870-5-2-95 (FT3) КП

Руководство Пользователя/1-е изд.

© 2014-2017. НПФ «КРУГ». Все права защищены.

Никакая часть настоящего издания ни в каких целях не может быть воспроизведена в какой бы то ни было форме и какими бы то ни было средствами, будь то электронные или механические, включая фотографирование, магнитную запись или иные средства копирования или сохранения информации, без письменного разрешения владельцев авторских прав.

Все упомянутые в данном издании товарные знаки и зарегистрированные товарные знаки принадлежат своим законным владельцам.

ООО НПФ «КРУГ»

РОССИЯ, 440028, г. Пенза, ул. Титова 1

Тел.: +7 (8412) 49-97-75, 49-94-14

Факс: +7 (8412) 55-64-96

E-mail: krug@krug2000.ru

<http://www.krug2000.ru>

Вы можете связаться со службой технической поддержки по E-mail:

support@krug2000.ru



СОДЕРЖАНИЕ

	Стр.
1 ОБЩИЕ СВЕДЕНИЯ	3
2 ОПИСАНИЕ ФАЙЛА КОНФИГУРАЦИИ ДРАЙВЕРА	4
2.1 Секция общих параметров [General Options].	4
2.2 Секция параметров для канала связи [Options ChannelX].	4
2.3 Секция параметров последовательного физического интерфейса канала связи [ChannelX serial].	5
2.4 Секция параметров устройства на канале связи [Options USOY ChannelX]	7
2.5 Секция описания передаваемых/принимаемых данных, участвующие в обмене с устройством [Attach USOY ChannelX]	8
3 СООБЩЕНИЯ ОБ ОШИБКАХ И КОДЫ ОШИБОК	13
4 ПРИМЕР ФАЙЛА КОНФИГУРАЦИИ ДРАЙВЕРА conf_uso.ini.	14

1 ОБЩИЕ СВЕДЕНИЯ

Драйвер протокола МЭК 870-5-2-95 (FT3) КП предназначен для организации информационного обмена по протоколу ГОСТ Р МЭК 60870-5-2-95 (формат кадра FT3).

Драйвер протокола FT3 обеспечивает сохранение истории изменения данных при пропадании связи и передачу их на верхний уровень при восстановлении связи.

Для этого драйвер протокола FT3 выделяет область памяти (буфер), куда записываются все события и блоки событий (данные класса 1 и 2), необходимые для передачи.

При отсутствии связи события сохраняются в буфер. В случае отсутствия связи и заполнения буфера событий класса, новые события заменяют самые старые события в буфере. При восстановлении связи при запросах событий данные передаются из буфера по старшинству, начиная с самых старых событий. При этом новые события также добавляются в буфер. Переданные события из буфера удаляются.

Все необходимые параметры работы драйвера задаются в файле конфигурации **conf_uso.ini**.

2 ОПИСАНИЕ ФАЙЛА КОНФИГУРАЦИИ ДРАЙВЕРА

Файл **conf_uso.ini** – это текстовый файл, который должен находиться в каталоге загружаемой БД контроллера **/gsw/settings** и иметь следующие секции:

- Секция общих параметров.
- Секция параметров для канала связи.
- Секция параметров последовательного физического интерфейса канала связи.
- Секция описания передаваемых/принимаемых данных, участвующие в обмене с устройством.

Все нижеописанные параметры настроек являются обязательными.
Строка комментария должна начинаться с символа «:» (двоеточие).

2.1 Секция общих параметров [General Options].

Секция **[General Options]** содержит описание параметров настроек общих для всех каналов связи данного контроллера.

Данная секция содержит следующие поля:

- **quan_channels** = *целочисленное значение*
Данное поле определяет количество каналов связи на данном контроллере.
Данное поле является обязательным.
Под каналами связи понимаются физические интерфейсы, обслуживаемые драйвером удаленных устройств работающие в CPB контроллера.

2.2 Секция параметров для канала связи [Options ChannelX].

Секция **[Options ChannelX]** содержит описание параметров настроек для канала связи с номером **X**.

X может принимать значения от 1 до **quan_channels** (см. описание секции **[General Options]**).

Данная секция содержит следующие поля:

- **type_connect** = *строковое значение*
Данное поле определяет тип соединения, которое используется при передаче данных.
Данное поле может принимать следующие значения:
Serial – передача данных происходит по последовательному физическому интерфейсу.
Данное поле является обязательным для работы драйвера.
- **type_protocol** = *строковое значение*
Данное поле определяет тип протокола, которое используется при передаче данных.
Данное поле может принимать следующие значения:
IEC_870_5_2_95_FT3 – передача данных происходит по протоколу ГОСТ Р МЭК 60870-5-2-95 (формат кадра FT3).
Данное поле является обязательным для работы драйвера.

- **type_USO**=строковое значение
Данное поле определяет тип удаленного устройства, с которым осуществляется передача данных. Данное поле может принимать следующие значения:
FT3 – передача данных происходит по протоколу ГОСТ Р МЭК 60870-5-2-95 (формат кадра FT3).
Данное поле является обязательным для работы драйвера.
- **quan_USO**=целочисленное значение
Данное поле определяет количество удаленных устройств подключенных к каналу связи с номером **X**.
Допустимое значение равно 1.
Данное поле является обязательным для работы драйвера.

2.3 Секция параметров последовательного физического интерфейса канала связи [ChannelX serial].

Секция [ChannelX serial] содержит описание параметров настроек последовательного физического интерфейса для канала связи с номером **X**.

X может принимать значения от 1 до **quan_channels** (см. описание секции [General Options]).

Данная секция содержит следующие поля:

- **com_number**=целочисленное значение
Данное поле определяет номер стандартного COM порта.
Диапазон значений 1- 6.
Данное поле является обязательным для работы драйвера.
- **485_controller**=строковое значение
Данное поле определяет тип платы последовательного интерфейса. Данное поле требуется для специфических типов.
Данное поле может принимать следующие значения:
none – стандартный последовательный интерфейс.
MDM-K – информационный обмен осуществляется через модем тональной частоты «МДМ-К». При отправке данных по протоколу FT3 через модем тональной частоты «МДМ-К» драйвер должен производить следующие действия:
– Перед отправкой данных драйвер должен выставить в единицу («1») сигнал RTS (Запрос передачи).
– Через временную задержку (от 0 до 255 мс) выдать информацию по протоколу FT3.
– Через временную задержку (от 0 до 255 мс) выставить в ноль («0») сигнал RTS.
Временные задержки задаются с параметров **com_RTS_on_delay** и **com_RTS_off_delay**.

В случае отсутствия данного поля берется значение по умолчанию равное **none**.

- **com_RTS_on_delay**=целочисленное значение
Данное поле определяет временную задержку при осуществлении информационного обмена через модем тональной частоты «МДМ-К». Значение поля определяет в мс задержку при отправке данных по протоколу FT3 между выставлением в единицу («1») сигнала RTS (Запрос передачи) отправкой пакета FT3. Выставление в единицу («1») сигнала RTS производится перед отправкой пакета FT3.

Данное поле может принимать значения от 0 до 255.

В случае отсутствия данного поля берется значение по умолчанию равное 0.

- **com_RTS_off_delay**=целочисленное значение

Данное поле определяет временную задержку при осуществлении информационного обмена через модем тональной частоты «МДМ-К». Значение поля определяет в мс задержку при отправке данных по протоколу FT3 между отправкой пакета FT3 b выставлением в ноль («0») сигнала RTS. Выставление в ноль («0») сигнала RTS производится после отправки пакета FT3.

Данное поле может принимать значения от 0 до 255.

В случае отсутствия данного поля берется значение по умолчанию равное 0.

- **data_flow**=строковое значение

Данное поле определяет режим обмена данными.

Данное поле может принимать следующие значения:

HD – полу дуплекс (Half Duplex).

FD – полный дуплекс (Full Duplex).

MS – мульти-точка (Multydrop-Slave) (приёмник всегда на линии, даже во время передачи). Данный флаг используется при наличии «эха» в канале связи.

В случае отсутствия данного поля берется значение по умолчанию равное **HD**.

Данное поле является обязательным для работы драйвера.

- **com_baud**=целочисленное значение

Данное поле определяет скорость обмена по последовательному интерфейсу. Скорость задается в бодах.

Данное поле может принимать значения: 300, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200.

Данное поле является обязательным для работы драйвера.

- **com_databits**=целочисленное значение

Данное поле определяет количество бит данных в каждом символе, передаваемом по последовательному интерфейсу.

Данное поле может принимать значения с 5 до 8.

- **com_stopbits**=целочисленное значение

Данное поле определяет количество стоп-битов в каждом символе, передаваемом по последовательному интерфейсу.

Данное поле может принимать значения 1 или 2.

- **com_parity**=строковое значение

Данное поле определяет режим контроля четности последовательного интерфейса.

Данное поле может принимать следующие значения:

not – режим контроля четности отключен.

even – генерируется и проверяется четное количество единичных битов символа посылки и бита контроля четности.

odd – генерируется и проверяется нечетное количество единичных битов символа посылки и бита контроля четности.

mark – генерируется и проверяется значение бита контроля четности, равное 1.

space – генерируется и проверяется значение бита контроля четности, равное 0.

Данное поле не является обязательным. В случае отсутствия данного поля берется значение по умолчанию равное **not**.

2.4 Секция параметров устройства на канале связи [Options USYO ChannelX]

Секция [Options USYO ChannelX] содержит описание параметров удаленного устройства с номером **Y** подсоединенного к каналу связи с номером **X**.

X может принимать значения от 1 до **quan_channels** (см. описание секции [General Options]).

Y может принимать значения от 1 до **quan_USO** (см. описание секции [Options ChannelX]).

Данная секция содержит следующие поля:

- **addressUSO=целочисленное значение**
Данное поле определяет сетевой адрес удаленного устройства **Y** подключенного к каналу связи с номером **X**.
Данное поле является обязательным для работы драйвера.
Данное поле определяет адрес (номер контроллера) **от 0 до 63**.
Рекомендации: данное значение определяется в зависимости от настроек сетевого адреса удаленного устройства. Удаленные устройства не могут иметь одинаковые сетевые адреса, если они подключены к одной сети.
- **Event_buffer_depth_c1=целочисленное значение**
Данное поле определяет количество событий, хранимых в буфере для событий класса 1 для канала связи с номером **X**.
Диапазон допустимых значений: **от 100 до 10000**.
Данное поле не является обязательным. В случае отсутствия данного поля берется значение по умолчанию равное **4000**.
- **Event_buffer_depth_c2=целочисленное значение**
Данное поле определяет количество событий, хранимых в буфере для событий класса 2 для канала связи с номером **X**.
Диапазон допустимых значений: **от 100 до 10000**.
Данное поле не является обязательным. В случае отсутствия данного поля берется значение по умолчанию равное **4000**.
- **var_statusUSO= строковое значение**
Данное поле определяет тип и номер переменной БД, посредством которой контролируется состояние связи с удаленным устройством (есть связь/нет связи).
В случае если есть связь с удаленным устройством, то эта переменная будет иметь значение равное **1**, иначе **0**.
Строковое значение имеет формат:
TTNNNN, где
TT – тип переменной,
NNNN – номер переменной в базе. Для данного поля допускаются следующие типы переменных:
ВД – входная дискретная.
Данное поле используется, если требуется контролировать состояние связи с удаленным устройством.
- **time_statusUSO=целочисленное значение**
Данное поле определяет время в секундах, на которое исключается сброс переменной (установка в «0»), определённой в поле **statusUSO**.
Диапазон допустимых значений: от 0 до 6000.
Данное поле не является обязательным. В случае отсутствия данного поля берется значение по умолчанию равное 60.

 Внимание!

Необходимо настроить адрес удаленного устройства согласно документации изготовителя.

2.5 Секция описания передаваемых/принимаемых данных, участвующие в обмене с устройством [Attach USOY ChannelX]

Секция **[Attach USOY ChannelX]** содержит описание привязок переменных БД контроллера к параметрам удаленного устройства с номером **Y** подсоединенного к каналу связи с номером **X**.

X может принимать значения от 1 до **quan_channels** (см. описание секции **[General Options]**).

Y может принимать значение 1.

В данной секции поля имеют следующий формат:

**<тип переменной БД><номер переменной БД>.а<номер атрибута> =
строковое значение**

, где

- **<тип переменной БД>** – тип переменной базы данных контроллера:
ВА – входная аналоговая;
АВ – аналоговая выходная;
ВД – входная дискретная;
ДВ – дискретная выходная;
РВ – ручной ввод.
- **<номер переменной БД>** – это порядковый номер переменной в БД контроллера (нумерация переменных начинается с 1)
- **<номер атрибута>** – это порядковый номер атрибута переменной в БД контроллера (нумерация атрибутов начинается с 1)
- **.а<номер атрибута>** – является необязательным параметром.



Внимание!

Все переменные, перечисленные в данной секции, должны иметь значение атрибута «номер платы» больше 200.
 Переменные типа РВ могут быть использованы только для привязки параметров строкового типа.

- Строковое значение имеет следующий формат:
**<номер группы объектов>_<номер объекта>_<номер параметра объекта>,
 <класс данных>,<тип параметра>,<описание типа данных параметра>**
 , где
 - **<номер группы объектов>_<номер объекта>_<номер параметра объекта>** – эти значения адресуют параметр
 - **<класс данных>** – номер класса данных, к которому относится значение переменной. Данное поле может принимать значения:
с0 – класс 0 (параметры, по которым не формируются события)
с1 – класс 1
с2 – класс 2.
 При отсутствии данного поля используется значение по умолчанию **с0**.
 - **<тип параметра>** – тип параметра:
s0 – тип 0 (байт состояния в ответе опускается);
s1 – тип 1 (байт состояния добавляется в ответ на запрос параметра данного типа);
s2 – тип 2 (байт состояния равный 0 – для контроллера).
 При отсутствии данного поля используется значение по умолчанию **s0**.
 - **<описание типа данных параметра>** имеет следующий формат:
 1) Вариант 1 – простой тип данных

Допустимые типы данных для параметров простого типа приведены в таблице 1:

Таблица 1 – Типы данных параметров простого типа

Наименование типа	Размер, байт	Описание
byte	1	Число (1 байт)
short	2	Число (2 байт)
float	4	Вещественное число
str16	16	Строка ASCII. Оканчивается символом конца строки.

Примеры привязок простого типа:

: Параметр 3.1.1 – ТТ1, Значение ФВ
AB101=3_1_1,s1,float

: Параметр 3.1.9 – ТТ1, Размерность
AB101=3_1_9,s2,byte

: Параметр 2.1.0 – ТУ1, Время выдержки ТУ
AB101=2_1_0,s2,short

2) Вариант 2 – массив, структура или массив структур одного типа:

(**<описание структуры данных >**), **<номер структуры в массиве>**,
<номер элемента в структуре >

, где

- (**<описание структуры данных>**) – перечень типов данных структуры данных через точку с запятой (“;”) в порядке их следования. Если тип данных массива состоит из одного элемента, то круглые скобки можно опустить. Допустимые типы данных приведены в таблице 2.

Таблица 2 – Типы данных параметров: массив, структура или массив структур одного типа

Условная группа	Наименование типа	Размер, байт	Описание
Простой тип	byte	1	Число (1 байт)
	short	2	Число (2 байт)
	float	4	Вещественное число
	str16	16	Строка ASCII. Оканчивается символом конца строки.
Структура «Дата-время»	dt_year	2	Год (0–99)
	dt_yday		День (1–366)
	dt_hour	1	Час (0–23)
	dt_dsec	2	1 единица = 100 мс

Формат (**<описание структуры данных>**) для структуры «Дата-время» (5 байт) будет выглядеть следующим образом:

(dt_year; dt_yday; dt_hour; dt_dsec)

Для сокращения описания структуры данных при наличии некоторого количества в структуре данных одинакового типа, можно использовать множитель их количества вместо простого их перечисления.

Примеры использования множителя количества идущих подряд в структуре данных одинакового типа:

1) группа 3, объект 1, параметр № 4 (min, max датчика) имеет формат 4 байта * 2 (т.е. 2 вещественных числа). В описании структуры данных такой формат можно описать следующими способами:

(float; float) или (float*2)

2) группа 4, объект 1, параметр № 11 (Среднее Q, dP, t, P, E за предыдущий час) имеет формат 1+4*5 (т.е. 1 байт и 4 вещественных числа). В описании структуры данных такой формат можно описать следующими способами:

(byte; float; float; float; float) или (byte; float*4)

3) сложный тип данных, где структура (byte;float*3;str16) повторяется 2 раза:

(byte; float*3; str16)*2

- **<номер структуры в массиве>** и **<номер элемента в структуре>** указывают конкретный элемент данных в структуре, привязываемый к указанной переменной БД. Данные описываются в порядке их следования. В простом случае, если нет в описании повторяющихся структур данных, то **<номер структуры в массиве>** равен **1**, **<номер элемента в структуре>** указывает порядковый номер привязываемого элемента структуры данных (нумерация начинается с **1**).

Примеры привязок значений параметров типа массив, структура или массив структур одного типа:

: 3_1_4 (min, max датчика) имеет формат 4*2 – (класс 1, тип 2)

: BA1 привязана к min датчика (на это указывает 1,1 в конце описания привязки)

BA1=3_1_4,c1,s2,(float*2),1,1

: BA2 привязана к max датчика (на это указывает 1,2 в конце описания привязки)

BA2=3_1_4,c1,s2,(float*2),1,2

: 4_1_11 (Среднее Q, dP, t, P, E за предыдущий час) имеет формат 1+4*5– (класс 2, тип 0)

: BA3 привязана к первому параметру (тип byte)

BA3=4_1_11,c2,s0,(byte;float*4),1,1

: BA4 привязана к параметру 2 (тип float)

BA4=4_1_11,c2,(byte;float*4),1,2

BA5=4_1_11,c2,(byte;float*4),1,3

BA6=4_1_11,c2,(byte;float*4),1,4

BA7=4_1_11,c2,(byte;float*4),1,5

Описание параметров процессорного модуля, к которым можно получить доступ по протоколу FT3, приведено в таблице 3.

Таблица 3 – Параметров процессорного модуля

Наименование	Формат данных по FT3	Доступ	Группа	Объект	№ параметра
Текущие дата-время	5 байт, дата-время	R/W	0	1	1

3 СООБЩЕНИЯ ОБ ОШИБКАХ И КОДЫ ОШИБОК

Условные обозначения:

%X – номер контроллера в сети

%N – код ошибки

%C – номер канала

1. “ЦП %X: DRV(ft3): Не найден файл конфигурации обмена”

Файл конфигурации обмена (“conf_uso.ini”) не найден в папке “/gsw/settings”.

2. “ ЦП %X: DRV(ft3): Ошибка конфигурации %N”

2 – Ошибка описания поля “[General Options] quan_channels”

3. “ ЦП %X: DRV(ft3): Канал%C Ошибка конфигурации %N”

N:

4 – Отсутствует поле “[Options Channel%C] type_connect”

5 – Отсутствует поле “[Options Channel%C] type_protocol”

6 – Отсутствует поле “[Options Channel%C] type_USO”

7 – Ошибка описания поля “[Options Channel%C] quan_USO”

8 – Ошибка описания поля “[Channel%C serial] com_number “

4. “ ЦП %X: DRV(ft3): Канал%C USO1 Ошибка конфигурации %N “

N:

10 – Ошибка описания поля “[Options USO1 Channel%C] addressUSO”.

11 – Указанная переменная не найдена в БД.

12 – Ошибка описания поля “[Options USO1 Channel%C] Event_buffer_depth_c1” или поля “[Options Channel%C] Event_buffer_depth_c2”.

13 – Ошибка описания поля “[Options USO1 Channel%C] time_statusUSO”.

5. “ ЦП %X: DRV(ft3): Канал%C Ошибка поля %s знач. по умолч.”

Используется значение по умолчанию для данного поля.

6. “ ЦП %X: DRV(ft3): Канал%C: Нет данных для обмена”

Нет ни одной (правильной) привязки параметров к БД СРВК.

7. “ ЦП %X: DRV(ft3): Канал%C USO1: Обмен ВКЛЮЧИЛ”

Обмен данными по каналу %C разрешён.

8. “ ЦП %X: DRV(ft3): Канал%C USO1: Обмен ВЫКЛЮЧИЛ”

Обмен данными по каналу %C запрещён.

9. “ ЦП %X: DRV(ft3): Остановлен.”

Драйвер остановлен вследствие критической ошибки.

4 ПРИМЕР ФАЙЛА КОНФИГУРАЦИИ ДРАЙВЕРА conf_uso.ini.

:Раздел общие настройки драйвера

[General Options]

quan_channels=1

:Раздел опций первого канала связи

[Options Channel1]

type_connect=Serial

type_protocol=IEC_870_5_2_95_FT3

type_USO=FT3

quan_USO=1

:Настройка соединения канала связи

[Channel1 serial]

com_number=1

485_controller=none

data_flow=FD

com_baud=115200

com_databits=8

com_stopbits=1

com_parity=even

[Options USO1 Channel1]

addressUSO=1

var_statusUSO=ВД2

time_statusUSO=600

Event_buffer_depth_c1=1000

Event_buffer_depth_c2=1000

[Attach USO1 Channel1]

: 3_1_4 (min, max датчика) имеет формат 4*2 – (класс 1, тип 2)

: BA10 привязана к min датчика

BA10=3_1_4,c1,s2,(float*2),1,1

: BA11 привязана к max датчика

BA11=3_1_4,c1,s2,(float*2),1,2

: 4_1_11 (Среднее Q, dP, t, P, E за предыдущий час) имеет формат 1+4*5– (класс 2, тип 0)

: BA21 привязана к первому параметру (тип byte)

BA21=4_1_11,c2,s0,(byte;float*4),1,1

: BA22 привязана к параметру 2 (тип float)

BA22=4_1_11,c2,(byte;float*4),1,2

BA23=4_1_11,c2,(byte;float*4),1,3

BA24=4_1_11,c2,(byte;float*4),1,4

BA25=4_1_11,c2,(byte;float*4),1,5

: байты 5–63 функции 01 – Read SuperFlo ID

BA41=14_1_6,(byte;str16;byte;str16;byte;str16;byte*7),1,1

BA42=14_1_6,(byte;str16;byte;str16;byte;str16;byte*7),1,2

BA43=14_1_6,(byte;str16;byte;str16;byte;str16;byte*7),1,3

BA44=14_1_6,(byte;str16;byte;str16;byte;str16;byte*7),1,4

BA45=14_1_6,(byte;str16;byte;str16;byte;str16;byte*7),1,5

BA46=14_1_6,(byte;str16;byte;str16;byte;str16;byte*7),1,6
BA47=14_1_6,(byte;str16;byte;str16;byte;str16;byte*7),1,7
BA48=14_1_6,(byte;str16;byte;str16;byte;str16;byte*7),1,8
BA49=14_1_6,(byte;str16;byte;str16;byte;str16;byte*7),1,9
BA50=14_1_6,(byte;str16;byte;str16;byte;str16;byte*7),1,10
BA51=14_1_6,(byte;str16;byte;str16;byte;str16;byte*7),1,11
BA52=14_1_6,(byte;str16;byte;str16;byte;str16;byte*7),1,12
BA53=14_1_6,(byte;str16;byte;str16;byte;str16;byte*7),1,13
BA54=14_1_6,(byte;str16;byte;str16;byte;str16;byte*7),1,14

: 4_1_12 Период

: Год

BA31=4_1_12,s2,(dt_year;dt_yday;dt_hour;short),1,1

:День

BA32=4_1_12,s2,(dt_year;dt_yday;dt_hour;short),1,2

:Час

BA33=4_1_12,s2,(dt_year;dt_yday;dt_hour;short),1,3

:кол-во часов

BA34=4_1_12,s2,(dt_year;dt_yday;dt_hour;short),1,4