Акционерное общество
«Инженерно-технический центр «Континуум»

ПРОГРАММНО-АППАРАТНЫЙ КОМПЛЕКС УСТРОЙСТВО СОПРЯЖЕНИЯ (Предобработчик 9-2)

Руководство программиста

КМБТ.108.002 Д1

Ярославль 2024

СОДЕРЖАНИЕ

Настоящий документ представляет собой версию 1.1 (от 31.01.2024 г.) руководства программиста на программно-аппаратный комплекс – устройство сопряжения и интеграции прецизионных широкополосных измерителей тока и напряжения с системой управления электропитанием сверхпроводящих магнитных катушек ИТЭР (Предобработчик).

Приведенные в документе сведения могут изменяться в связи с возможным выходом обновленных версий Предобработчика. Соответствующие изменения будут отражены в последующих версиях настоящего руководства.

Перед началом работы рекомендуется предварительно ознакомиться с основами функционирования Предобработчика. Основы функционирования Предобработчика, описаны в руководстве по эксплуатации (КМБТ.108.002 РЭ).

Перед началом работы рекомендуется предварительно ознакомиться с регламентами действия оператора с Предобработчиком. Регламент действий оператора Предобработчика, описаны в руководстве оператора (КМБТ.108.002 Д2).

1 Перечень использованных в документе сокращений и терминов

Перечень использованных сокращений:

ИТЭР (ITER)	–	International Thermonuclear Experimental Reactor (англ.) (проект международного экспериментального термоядерного реактора)
ОС	–	Операционная система
ПО	–	Программное обеспечение
APDU	–	Application Protocol Data Unit (eng., протокольный блок данных прикладного уровня (термин – в соотв. с IEC 61850-9-2))
API	–	Application Programming Interfaces (eng., «интерфейсы программирования приложений»)
AVG	–	«AVeraGe» (eng.), в данном документе под сокращением подразумевается характеристика «среднее (арифметическое) за период времени значение сигнала (тока или напряжения)»
CPU	–	«Central Processing Unit» (eng.), в терминологии FastController-ов ИТЭР – базовый вычислительный блок FastController-а
FCS	–	«Frame Check Sequence» (eng.), последовательность (бит) проверки фрейма (пакета) Ethernet (поле пакета Ethernet, содержащее строку бит, располагаемую в конце пакета, содержащее значение контрольной суммы данных пакета)
PXIe	–	«PCI eXtension for Instrumentation» (eng.), стандарт модульного измерительного оборудования
RMS	–	«Root-Mean Square» (eng., «среднеквадратическое значение»), величина действующего (среднеквадратического) значения сигнала тока или напряжения
SOF	–	«Start-Of-Frame Delimeter» (eng.), разделитель начала кадра (битовое поле пакета Ethernet, характеризующее начало пакета)
VLAN	–	«Virtual local area network» (eng.), виртуальная локальная сеть (в соотв. с IEEE 802.1Q)

Перечень использованных терминов:

Комплекс измерений электрических параметров системы управления электропитанием сверхпроводящих магнитных катушек ИТЭР – Комплекс, включающий в себя совокупность высоковольтных измерительных датчиков напряжения и тока с цифровым выходом, обеспечивающих выполнение измерений тока и напряжения с преобразованием данных измерений в цифровой вид в системе электропитания сверхпроводящих магнитных катушек ИТЭР. Вышеуказанный комплекс обеспечивает предоставление необходимых данных измерений токов и напряжений в цифровом виде в систему управления электропитанием сверхпроводящих магнитных катушек ИТЭР.

Комплекс обработки данных (FastController) системы управления электропитанием сверхпроводящих магнитных катушек ИТЭР – Вычислительный комплекс, обеспечивающий в качестве одной из своих задач обработку данных измерений с датчиков тока и напряжения из состава комплекса измерений электрических параметров системы управления электропитанием сверхпроводящих магнитных катушек ИТЭР (см. выше) с целью обеспечения функции оперативного управления электропитанием сверхпроводящих магнитных катушек ИТЭР

Поток 9-2 – Упорядоченная последовательность Ethernet-пакетов, транслируемая между узлом-приемником и узлом-получателем в сети Ethernet, соответствующая стандарту IEC 61850-9-2 (либо спецификации IEC 61850-9-2 LE), характеризуемая уникальным идентификатором последовательности, содержащимся в поле MsvID/UsvID каждого из Ethernet-пакетов указанной последовательности

2 Общие сведения

2.1 Назначение

2.1.1 Предобработчик является встраиваемым компонентом, подключаемым последовательно в тракт, соединяющий FastController, либо иной персональный компьютер (ПК) со специализированным программным обеспечением обработчик данных, с датчиками тока и напряжения с цифровым выходом формата IEC 61850-9-2.

2.1.2 Основной функцией Предобработчика является приём и предварительная обработка цифровых коммуникационных потоков от измерительных датчиков с последующим предоставлением результатов этой обработки приложениям, выполняющимся на платформе персонального компьютера.

2.1.3 Предобработчик оснащается следующим набором специализированных интерфейсов:

– внешние коммуникационные интерфейсы (функциональная специализация – захват цифровых коммуникационных потоков в формате IEC 61850-9-2 из коммуникационной сети);

– внутренний коммуникационный интерфейс (функциональная специализация – взаимодействие с FastController).

2.1.4 Предобработчик обеспечивает обработку данных внешних цифровых потоков в формате IEC 61850-9-2, подаваемых на входы внешних коммуникационных интерфейсов Ethernet устройства, с передачей результатов обработки данных через внутренний коммуникационный интерфейс (PXIe) устройства на персональный компьютер.

2.1.5 Обработка данных, выполняемая Предобработчиком, включает выделение из принимаемых устройством цифровых потоков в формате IEC 61850-9-2 только «полезных» данных (в частности, только последовательностей мгновенных значений тока и/или напряжения с привязкой ко времени) и обработку «служебной» информации на базе собственных вычислительных ресурсов. При этом обеспечивается сокращение количества данных, передаваемых на ПК, а также обеспечивается разгрузка вычислительных ресурсов ПК вследствие исключения необходимости выполнения непосредственно в ПК вышеуказанной обработки данных.

2.1.6 Кроме того, в рамках обработки данных обеспечивается выполнение арифметических операций над данными (мгновенными значениями электрических величин) принимаемых цифровых потоков в формате IEC 61850-9-2 с передачей результатов выполнения указанных операций на ПК. В частности, обеспечивается формирование выходного «дифференциального» потока (т.е. потока, мгновенные значения которого являются разностью мгновенных значений двух цифровых потоков в формате IEC 61850-9-2), выполнение свёртки массивов мгновенных значений

(вычисление среднего (арифметического) значения сигнала тока / напряжения для соответствующего массива мгновенных значений) и др.

2.2 Состав программно-технических средств и окружения

2.2.1 В состав Предобработчика входит следующий набор программно-аппаратных средств:

- Предобработчик, включающий в себя аппаратную часть в виде платы сопряжения (PXIe) и встраиваемое программное обеспечение (ВПО), функционирующее в составе вышеуказанного Предобработчика;
- комплект программного обеспечения (ПО) интеграции Предобработчика с операционной системой Astra Linux (ОС Linux), включающий:
  1) драйвер платы сопряжения для операционной системы Astra Linux;
  2) программную библиотеку (предоставляющую функции API) для доступа к модулю сопряжения из программных приложений ОС Linux.

2.2.2 Состав компонентов Предобработчика и его окружения, а также их размещение в программно-аппаратных комплексах показаны на рисунке 1.

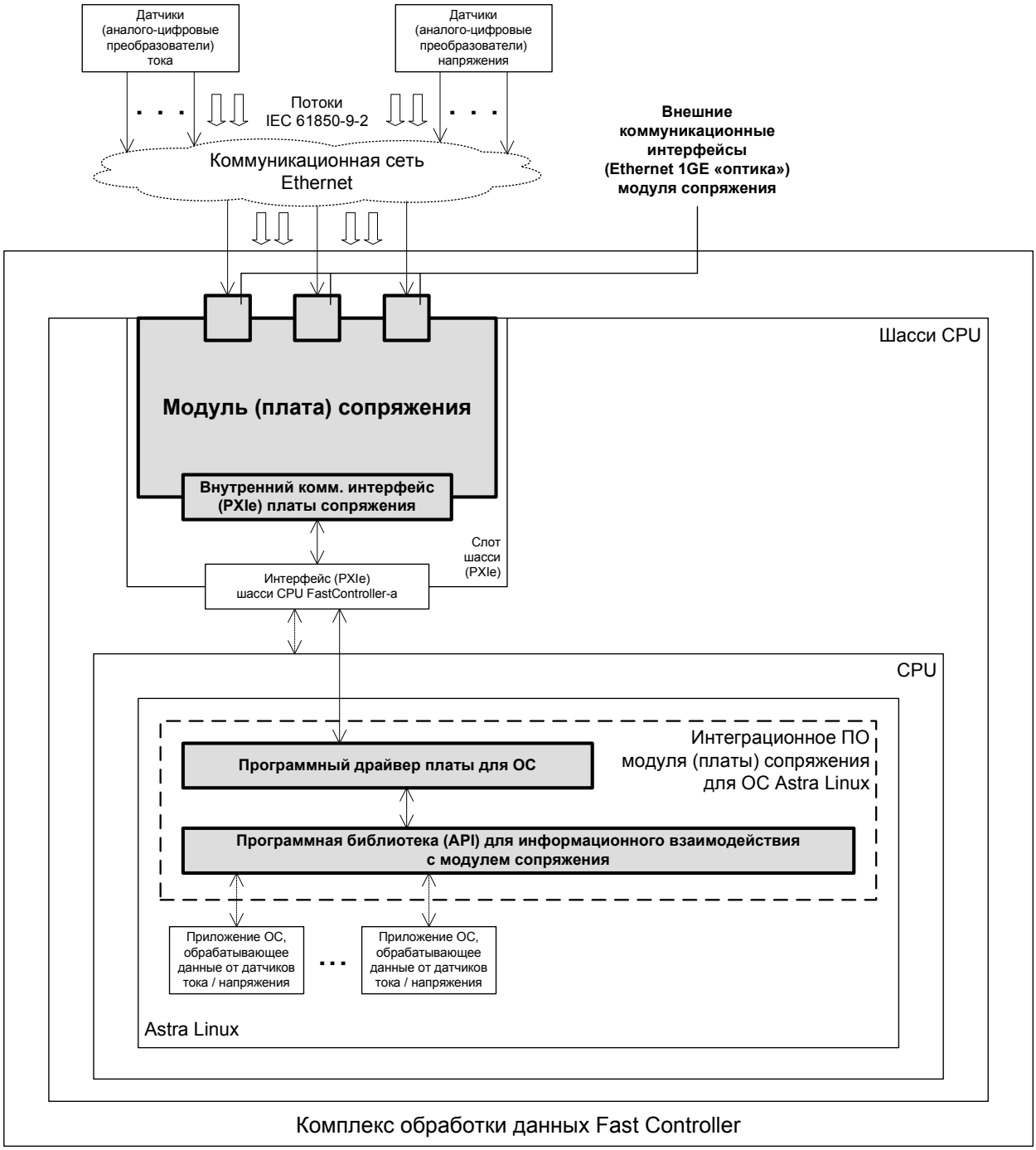

Рисунок 1 – Состав компонентов Предобработчика и его окружения, а также их размещение в программно-аппаратных комплексах

2.3 Описание Предобработчика

2.3.1 Предобработчик, включающий в себя плату и встраиваемое ПО, обеспечивает обработку данных внешних потоков 9-2, подаваемых на входы внешних коммуникационных интерфейсов Ethernet Предобработчика, с передачей результатов обработки данных через внутренний коммуникационный интерфейс платы на ПК.

2.3.2 В рамках вышеуказанной обработки данных обеспечивается, в частности, выделение из принимаемых Предобработчиком потоков 9-2

только «полезных» данных (в частности, только последовательностей мгновенных значений тока или напряжения с привязкой ко времени) с отбрасыванием «служебной» информации. При этом обеспечивается сокращение более чем на порядок количества данных, передаваемых на ПК, а также разгрузка вычислительных ресурсов ПК вследствие исключения необходимости выполнения непосредственно в ПК вышеуказанной обработки данных.

2.3.3 Кроме того, в рамках вышеуказанной обработки данных обеспечивается выполнение арифметических операций над данными (мгновенные значения электрических величин) принимаемых потоков 9-2 с передачей результатов выполнения операций на FastController. В частности, обеспечивается формирование выходного «дифференциального» потока (т.е. потока, мгновенные значения которого являются разностью мгновенных значений двух потоков 9-2), выполнение свертки массивов мгновенных значений (напр., вычисление RMS сигнала для соответствующего массива мгновенных значений сигнала тока или напряжения) и т.п.

2.4 Описание драйверов ОС Astra Linux

2.4.1 Драйвер Предобработчика предназначен для обеспечения функционирования Предобработчика, установленного в ПК, в программной среде ОС Astra Linux.

2.4.2 Наличие указанного драйвера, установленного в ОС Astra Linux ПК, является обязательным для обеспечения функционирования модуля сопряжения в составе ПК.

2.4.3 Указанный программный драйвер Предобработчика для ОС Astra Linux входит в поставку и поставляется на компакт-диске, либо ином электронном носителе.

2.5 Описание программной библиотеки (API)

2.5.1 Программная библиотека, предоставляющая функции API для доступа к Предобработчику из ОС Astra Linux, предоставляет прикладному ПО, функционирующему в среде ОС Astra Linux, сервисы (API) для информационного взаимодействия с Предобработчиками.

2.5.2 Для всех Предобработчиков, одновременно подключенных к ПК (возможно подключение одновременно до 16-ти Предобработчиков в одно шасси PXIe) используется один и тот же экземпляр программной библиотеки (API), установленной в ОС Astra Linux.

2.5.3 Указанная программная библиотека (API) входит в поставку Предобработчиков и поставляется на компакт-диске либо, ином электронном носителе.

2.6 Сведения о настоящем руководстве

2.6.1 Настоящий документ описывает высокоуровневые механизмы информационного взаимодействия с Предобработчиком, предоставляемые программным приложениям ОС Astra Linux программной библиотекой (API) из состава поставки Предобработчика.

Примечание: Работа с Предобработчиком посредством низкоуровневых сервисов ввода-вывода, предоставляемых программным драйвером для ОС Linux, требует особой внимательности, т.к. использование указанных сервисов при неосторожном использовании может привести к последующей некорректной работе, а в отдельных случаях, и к полной неработоспособности Предобработчика.

2.6.2 Приведенные в настоящем руководстве сведения предназначены для использования программистами, разрабатывающими программные приложения для ОС Astra Linux, предназначенные для выполнения в рамках ПК для обработки данных внешних потоков 9-2, получаемых от внешних датчиков тока и напряжения с цифровым выходом МЭК 61850-9-2.

2.6.3 Настоящее руководство содержит в себе описание вызовов функций API, реализуемых программной библиотекой (API), а также описание дополнительных структур и типов данных, используемых программной библиотекой (API).

3 Технические характеристики

3.1 Конструктивные характеристики Предобработчика

3.1.1 Предобработчик выполнен на основе электронной платы в форм-факторе PXIe 3U, устанавливаемой в объединительную плату расширения шасси PXIe.

3.1.2 Предобработчик выполнен с внутренним коммуникационным интерфейсом PXIe – для связи (сопряжения) с ПК.

3.1.3 Предобработчик содержит в себе три внешних коммуникационных интерфейса Ethernet 1 Гбит/с («оптика») для подключения к цифровым выходам первичных датчиков измерения тока и напряжения (IEC 61850-9-2). Все три указанных интерфейса Ethernet соответствуют спецификации 1000BASE-SX и выполнены с разъемами для подключения («оптика») типа LC (в соотв. с IEC 61754-20).

3.1.4 Габариты, занимаемые платой в корпусе шасси – одно штатное место для установки одной платы расширения PXIe.

3.2 Характеристики данных, принимаемых и транслируемых Предобработчика через внешние коммуникационные интерфейсы Ethernet

3.2.1 Характеристики принимаемых и поддерживаемых (распознаваемых) Предобработчиком входных потоков 9-2 от измерительных датчиков тока и напряжения приведены ниже в таблице 1.

Таблица 1 – Характеристики поддерживаемых (распознаваемых) Предобработчиком цифровых потоков формате IEC 61850-9-2

№	Тип профиля	Частота дискретизации[1], Гц	Количество и тип мгновенных значений в каждом срезе потока	«noASDU» (параметр потока 9-2)[3]	Частота передачи фреймов, Гц[4]	Спецификация протокола передачи
1	«92LE»	12 800	4 значения тока[2] + + 4 значения напряжения[2]	8	1600	IEC 61850-9-2LE

№	Тип профиля	Частота дискретизации[1], Гц	Количество и тип мгновенных значений в каждом срезе потока	«noASDU» (параметр потока 9-2)[3]	Частота передачи фреймов, Гц[4]	Спецификация протокола передачи
2	«HVDC»	100 000	1 значение напряжения	1	100 000	IEC 61850-9-2 (Ed.1.0/2.0)– модифицированная спецификация[5]

[1] Количество срезов мгновенных значений исходного аналогового сигнала измеряемой величины (тока или напряжения) в секунду

[2] 3 фазы + нейтраль

[3] Параметр цифрового потока в формате IEC 61850-9-2, представляющий собой количество срезов мгновенных значений, содержащихся в одном Ethernet-фрейме потока. Значение указанного параметра содержится в каждом из Ethernet-фреймов цифрового потока в формате IEC 61850-9-2 в соответствующем поле фрейма.

[4] Указана номинальная частота передачи (приема устройством сопряжения) Ethernet-фреймов цифрового потока в формате IEC 61850-9-2. Фактическая частота передачи фреймов может отличаться от номинальной частоты. Отличие частоты передачи фреймов от номинальной может быть вызвано, например, дополнительными задержками фреймов в Ethernet-коммутаторах или дополнительной коммуникационной нагрузкой помимо данного цифрового потока в формате IEC 61850-9-2 (в т. ч., возможно, нагрузкой от других цифровых потоков в формате IEC 61850-9-2) в сети Ethernet. Допустимое отклонение фактической частоты передачи Ethernet-фреймов цифрового потока в формате IEC 61850-9-2 соответствующего профиля от номинальной определяется допустимой задержкой при передаче каждого отдельного фрейма указанного потока через коммуникационную сеть Ethernet. При этом устройство сопряжения обеспечивает корректную обработку данных входных цифровых потоков в формате IEC 61850- 9-2, получаемых на вход внешних интерфейсов Ethernet устройства, при величине задержки отдельных фреймов в сети Ethernet до 10 миллисекунд.

[5] Характеристики модифицированной спецификации IEC 61850-9-2 приведены в Приложении А к данному РП .

3.2.2 Предобработчик игнорирует при приеме на вход каждого из трех своих внешних интерфейсов Ethernet трафик, отличный от трафика указанного типа. При этом игнорируются, в том числе, потоки 9-2, соответствующие иным типам профиля.

3.2.3 В процессе своего функционирования Предобработчик не формирует никакого выходного трафика через каждый из своих 3-х внешних коммуникационных интерфейсов Ethernet.

3.2.4 Размер Ethernet-фреймов, подаваемых на вход каждого из трех интерфейсов Ethernet Предобработчика, должен составлять не более 1522 байта (с учетом преамбулы, поля SOF и FCS).

Примечание: При нарушении указанного условия корректное функционирование Предобработчика не гарантируется.

3.3 Характеристики коммуникационно-вычислительной производительности Предобработчика

3.3.1 Предобработчик обеспечивает одновременную обработку данных до четырех входных потоков 9-2 типа «HVDC».

Примечание 1: Обработка данных включает в себя разбор принимаемых потоков, передачу данных из потоков через внутренний коммуникационный интерфейс Предобработчика в ПК, а также, возможно, выполнение промежуточных математических операций над данными принимаемых и разбираемых потоков 9-2.

Примечание 2: Указанные четыре входных потока могут быть распределены для приема по всем трем внешним интерфейсам Ethernet Предобработчика (напр., два потока принимаются на один интерфейс Ethernet, два других потока – по одному на два других интерфейса Ethernet модуля), либо быть все поданы на один и тот же (из трех) внешних интерфейсов Ethernet модуля.

3.3.2 Предобработчик обеспечивает одновременную обработку данных до двенадцати входных потоков 9-2 типа «92LE».

Примечание 1: Обработка данных включает в себя разбор принимаемых потоков, передачу данных из потоков через внутренний коммуникационный интерфейс Предобработчика в ПК, а также, возможно, выполнение промежуточных математических операций над данными принимаемых и разбираемых потоков 9-2.

Примечание 2: Указанные 12 входных потока могут быть распределены для приема по всем трем внешним интерфейсам Ethernet Предобработчика (напр., шесть потоков принимаются на один интерфейс Ethernet, шесть других потоков – по три на два других интерфейса Ethernet Предобработчика), либо быть все поданы на один и тот же (из трех) внешних интерфейсов Ethernet.

3.3.3 Предобработчик обеспечивает корректную обработку данных входных потоков 9-2 при максимальной коммуникационной загрузке входящим трафиком (т.е. до 1 Гбит/с) одновременно всех трех внешних коммуникационных интерфейсов Ethernet.

Примечание: Данное ограничение выражает допустимый объем входного трафика по протоколу МЭК 61850-9-2, подаваемого на вход каждого из трех интерфейсов Ethernet, что исключает коммуникационо-вычислительную перегрузку Предобработчика входным трафиком по протоколу МЭК 61850-9-2. В данном случае, под входным трафиком по протоколу МЭК 61850-9-2 понимаются серии передаваемых по каналу Ethernet-фреймов, в теле каждого из которых в поле LENGTH/TYPE (в случае нетэгированного трафика, т.е. Ethernet-фреймы без префикса «QTag» согласно IEEE 802.3), либо в поле «MAC CLIENT LENGTH/TYPE» (в случае тэгированного трафика, т.е. при наличии префикса «Qtag» в теле

Ethernet-фрейма) содержится значение 0x88BA (HEX), что соответствует фрейму потока 9-2 согласно спецификации IEC 61850-9-2. При этом допустимая частота передачи указанных фреймов на вход каждого из трех внешних интерфейсов Ethernet модуля сопряжения составляет не более 400 000 фреймов в секунду. При превышении указанной величины не гарантируется корректное функционирование модуля сопряжения ввиду чрезмерной коммуникационно-вычислительной перегрузки модуля, вызванной чрезмерным входным трафиком 9-2.

4 Описание структур и типов данных, используемых библиотекой (API)

4.1 Перечень констант и типов данных

4.1.1 Тип потока данных является целочисленной величиной от 0 до 2. Для данного типа величин определены следующие мнемонические обозначения:

– **PP92_FLOW_PROFILE_DISABLED** = 0, для отключенного потока;

– **PP92_FLOW_PROFILE_92LE** = 1, для потока типа «92LE»;

– **PP92_FLOW_PROFILE_HVDC** = 2, для потока типа «HVDC».

4.1.2 Тип измерительного канала является целочисленной величиной от 0 до 2. Для данного типа величин определены следующие мнемонические обозначения:

– **PP92_CHANNEL_FORMAT_DISABLED** = 0, для отключенного измерительного канала;

– **PP92_CHANNEL_FORMAT_FLOAT_LE** = 1, для измерительного канала в котором данные представлены в соответствии со стандартом IEEE 754 для значений в формате «от младшего к старшему»;

– **PP92_CHANNEL_FORMAT_FLOAT_BE** = 2, для измерительного канала в котором данные представлены в соответствии со стандартом IEEE 754 для значений в формате «от старшего к младшему».

4.1.3 Максимально допустимая длина идентификатора потока SvID равна 32. Для данной константы определено следующее мнемоническое обозначение – **PP92_FLOW_SVID_MAX_LEN**.

4.1.4 Максимально допустимая длина арифметического выражения в конфигурации измерительного канала равна 256. Для данной константы определено следующее мнемоническое обозначение – **PP92_CHANNEL_EXPR_LEN_MAX**.

4.2 Структуры данных

4.2.1 Структура данных **pp92_ident** используется для идентификации Предобработчиков и включает в себя следующие элементы:

– **serial_number** – тип данных беззнаковое 32-х битное целочисленное, для хранения серийного номера Предобработчика;

– **sw_version_major** – тип данных беззнаковое 16-х битное целочисленное, для хранения основной части номера версии программного обеспечения;

– **sw_version_minor** – тип данных беззнаковое 16-х битное целочисленное, для хранения дополнительной части номера версии программного обеспечения;

– **hw_version_major** – тип данных беззнаковое 16-х битное целочисленное, для хранения основной части номера версии аппаратной реализации;

– **hw_version_minor** – тип данных беззнаковое 16-х битное целочисленное, для хранения дополнительной части номера версии аппаратной реализации.

4.2.2 Структура данных **pp92_health** используется для хранения данных состояния Предобработчиков и включает в себя следующие элементы:

– **uptime** – тип данных беззнаковое 32-х битное целочисленное, для хранения количества секунд, прошедших с момента включения Предобработчика;

– **temperature** – тип данных беззнаковое 32-х битное целочисленное, для хранения текущей температуры процессора Предобработчика (в миллиградусах Цельсия).

4.2.3 Структура данных **pp92_flow_config** используется для хранения конфигурации потока данных и включает в себя следующие элементы:

– **profile** – тип данных беззнаковое 32-х битное целочисленное, для хранения типа потока (**PP92_FLOW_PROFILE_***);

– **svid_len** – тип данных беззнаковое 32-х битное целочисленное, для хранения актуальной длины идентификатора потока;

– **svid** – массив беззнаковых 8-х битных целочисленных чисел, для хранения кодов символов идентификатора потока. Размер массива ограничен максимальной длиной **PP92_FLOW_SVID_MAX_LEN**. В качестве значений допускаются арабские цифры и буквы латинского алфавита.

4.2.4 Структура данных **pp92_flow_filter** используется для хранения данных для фильтрации потоков данных и включает в себя следующие элементы:

– **port** – тип данных беззнаковое 32-х битное целочисленное, для хранения номера Ethernet порта Предобработчика для ограничения;

– **dst_mac** – массив беззнаковых 8-х битных целочисленных чисел, для хранения кодов символов MAC адреса источника данных 9-2. Размер массива ограничен 6. В качестве значений допускаются арабские цифры и буквы латинского алфавита ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’;

– **src_mac** – массив беззнаковых 8-х битных целочисленных чисел, для хранения кодов символов MAC адреса приемника данных 9-2. Размер массива ограничен 6. В качестве значений допускаются арабские цифры и буквы латинского алфавита ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’;

– **vlanid** – тип данных беззнаковое 32-х битное целочисленное, для хранения номера VLAN идентификатора.

4.2.5 Структура данных **pp92_flow_info** используется для хранения конфигурационных данных потока и включает в себя следующие элементы:

– **port** – тип данных беззнаковое 32-х битное целочисленное, для хранения номера Ethernet порта Предобработчика для ограничения;

– **dst_mac** – массив беззнаковых 8-х битных целочисленных чисел, для хранения кодов символов MAC адреса источника данных 9-2. Размер массива ограничен 6. В качестве значений допускаются арабские цифры и буквы латинского алфавита ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’;

src_mac – массив беззнаковых 8-х битных целочисленных чисел, для хранения кодов символов MAC адреса приемника данных 9-2. Размер массива ограничен 6. В качестве значений допускаются арабские цифры и буквы латинского алфавита ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’;

– **vlanid** – тип данных беззнаковое 32-х битное целочисленное, для хранения номера VLAN идентификатора;

– **appid** – тип данных беззнаковое 32-х битное целочисленное, для хранения номера значения поля APPID протокола IEC 61850-9-2 (тип протокола);

– **confrev** – тип данных беззнаковое 32-х битное целочисленное, для хранения номера значения поля ConfRev протокола IEC 61850-9-2 (номер ревизии конфигурации источника данных IEC 61850-9-2);

– **smpsynch** – тип данных беззнаковое 32-х битное целочисленное, для хранения номера значения поля SmpSync протокола IEC 61850-9-2 (состояние синхронизации источника IEC 61850-9-2).

4.2.6 Структура данных **pp92_flow_stats** используется для хранения статистических данных потока и включает в себя следующие элементы:

– **received_asdu** – тип данных беззнаковое 64-х битное целочисленное, для хранения количества полученных блоков данных;

– **dropped_asdu** – тип данных беззнаковое 64-х битное целочисленное, для хранения количества пропущенных блоков данных;

– **unordered_asdu** – тип данных беззнаковое 64-х битное целочисленное, для хранения количества блоков данных, полученных с нарушением последовательности нумерации (SmpCnt).

4.2.7 Структура данных **pp92_channel_config** используется для хранения статистических данных потока и включает в себя следующие элементы:

– **format** – тип данных беззнаковое 32-х битное целочисленное, для хранения типа измерительного канала данных (PP92_CHANNEL_FORMAT_*);

– **block_size** – тип данных беззнаковое 32-х битное целочисленное, для хранения размера «окна» для усреднения в процессе вычисления интегральных значений;

– **expr_len** – тип данных беззнаковое 32-х битное целочисленное, для хранения длины арифметического выражения в измерительном канале;

– **expr** – массив беззнаковых 8-х битных целочисленных чисел, для хранения кодов символов идентификатора потока. Размер массива ограничен максимальной длиной **PP92_CHANNEL_EXPR_LEN_MAX**. Требования к

допустимому значению приведено в руководстве оператора КМБТ.108.002 Д2.

4.2.8 Структура данных **pp92_channel_stats** используется для хранения статистических данных измерительного канала и включает в себя следующие элементы:

- **queued_chunks** – тип данных беззнаковое 64-х битное целочисленное, для хранения количества обработанных блоков данных;
- **dropped_chunks** – тип данных беззнаковое 64-х битное целочисленное, для хранения количества пропущенных блоков данных.

4.2.9 Структура данных **pp92_port_status** используется для хранения данных статуса порта Ethernet Предобработчика и включает в себя следующие элементы:

- **link** – тип данных беззнаковое 32-х битное целочисленное, для хранения коммуникационного статуса порта Ethernet;
- **speed** – тип данных беззнаковое 32-х битное целочисленное, для хранения номинальной скорости установившегося Ethernet соединения;
- **duplex** – тип данных беззнаковое 32-х битное целочисленное, для хранения признака дуплексного установившегося Ethernet соединения.

4.2.10 Структура данных **pp92_port_stats** используется для хранения статистических данных по порту Ethernet Предобработчика и включает в себя следующие элементы:

- **received_packets** – тип данных беззнаковое 64-х битное целочисленное, для хранения количества полученных Ethernet пакетов;
- **dropped_packets** – тип данных беззнаковое 64-х битное целочисленное, для хранения количества пропущенных Ethernet пакетов;
- **errors_packets** – тип данных беззнаковое 64-х битное целочисленное, для хранения количества Ethernet пакетовю принятых с ошибками.

4.2.11 Структура данных **pp92_block** используется для хранения данных по блоку памяти, используемому для передачи в драйвер ОС Linux, и включает в себя следующие элементы:

- **quality** – тип данных беззнаковое 8-х битное целочисленное, для хранения признака качества данных на основании признаков качества протокола IEC 61850-9-2;
- **channel** – тип данных беззнаковое 8-х битное целочисленное, для хранения номера измерительного канала;
- **count** – тип данных беззнаковое 16-х битное целочисленное, для хранения количества мгновенных значений в блоке памяти;
- **counter** – тип данных беззнаковое 32-х битное целочисленное, для хранения порядкового номера первого мгновенного значения в блоке памяти;
- **samples** – тип данных беззнаковое 32-х битное целочисленное, для хранения величины смещения в блоке памяти для мгновенных значений;

– **aggreates** – тип данных беззнаковое 32-х битное целочисленное, для хранения величины смещения в блоке памяти для интегральных значений.

4.3 Блоки данных

4.3.1 Предобработчиком осуществляется прием и фильтрация входящего Ethernet трафика. Полученные и прошедшие фильтр данные подвергаются предварительной обработке, результатом которой является формирование блока данных, содержащего набор мгновенных значений и результаты вычислений интегральных параметров.

4.3.2 Блоки данных организованы в порядке очереди, т.е. данные поступающие от результатов обработки встают в начало очереди, а для API доступен блок памяти в конце очереди. Общий объем памяти на блоки данных ограничен.

4.3.3 Если приложение ОС через API не будет успевать выбирать блоки данных, то данные будут утеряны.

4.4 Коды ошибок

4.4.1 Функции, включенные в состав API, являются используют штатные механизмы доступа к ресурсам ОС Linux. В связи с этим, большинство кодов ошибок являются кодами ошибок ОС Linux при доступе к ресурсам.

4.4.2 Функция **pp92_close** предусматривает следующие коды ошибок:

– EINVAL – неверный аргумент (POSIX.1);
– ENODEV – нет такого устройства (POSIX.1).

4.4.3 Функции **pp92_get_ident, pp92_get_health, pp92_set_flow_config, pp92_get_flow_config, pp92_set_flow_filter, pp92_get_flow_filter, pp92_get_flow_info, pp92_get_flow_stats, pp92_set_channel_config, pp92_get_channel_stats, pp92_get_port_status, pp92_get_port_stats, pp92_stop, pp92_dequeue, pp92_enqueue, pp92_is_running** предусматривает следующие коды ошибок:

– EINVAL – неверный аргумент (POSIX.1);
– ENODEV – нет такого устройства (POSIX.1);
– EBADF – неправильный дескриптор файла (POSIX.1);
– EFAULT – неправильный адрес (POSIX.1);
– ENOTTY – неподходящая операция управления вводом/выводом (POSIX.1).

4.4.4 Функция **pp92_get_port_stats** предусматривает следующие коды ошибок:

– EINVAL – неверный аргумент (POSIX.1);
– ENODEV – нет такого устройства (POSIX.1);
– EACCES – доступ запрещён (POSIX.1);
– EAGAIN – ресурс временно недоступен;

– EBADF – неправильный дескриптор файла (POSIX.1)
– ENFILE – слишком много открытых файлов в системе (POSIX.1);
– ENOMEM – недостаточно места (POSIX.1);
– EPERM – операция не позволена (POSIX.1);
– EFAULT – неправильный адрес (POSIX.1);
– ENOTTY – неподходящая операция управления вводом/выводом (POSIX.1).

4.4.5 Функция **pp92_wait** предусматривает следующие коды ошибок:

– EINVAL – неверный аргумент (POSIX.1);
– ENODEV – нет такого устройства (POSIX.1);
– EAGAIN – ресурс временно недоступен;
– EFAULT – неправильный адрес (POSIX.1);
– EINTR – прерванный вызов функции (POSIX.1), вызов прерван обработчиком сигналов;
– ENOMEM – недостаточно места (POSIX.1).

5 Описание функций, предоставляемых библиотекой API

5.1 Функция «pp92_open»

5.1.1 Для выполнения различных операций с Предобработчиком из прикладного ПО, функционирующего в среде ОС Astra Linux, посредством программных вызовов, предоставляемых библиотекой (API) платы, включая:

- считывание с платы данных;
- конфигурирование платы;

необходимо выполнить функцию **pp92_open**. Функция аналогична открытию файла в операционной системе Linux.

Синтаксис функции **pp92_open**:

Pp92 **pp92_open** (const char *device),

где:

- device – имя Предобработчика в ОС Astra Linux;
- возвращаемое функцией значение – возвращаемый объект типа Pp92 для использования в дальнейших функциях.

5.2 Функция «pp92_close»

5.2.1 Для выполнения освобождения памяти, выделенной ранее при выполнении операции **pp92_open** необходимо выполнить функцию «pp92_close». Функция аналогична закрытию файла в операционной системе Astra Linux.

Синтаксис функции **pp92_close**:

int **pp92_close** (Pp92 pp),

где:

- pp – объект, полученный ранее при выполнении функции pp92_open;
- возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки, аналогичный коду ошибки при закрытии файла в ОС Astra Linux.

5.3 Функция «pp92_start»

5.3.1 Для начала обработки входящего трафика Предобработчику необходимо передать команду на начало работы **pp92_start**. До начала обработчик потока Предобработчик должен быть корректно сконфигурирован (настроены фильтры трафика, потоки данных и измерительные каналы).

Синтаксис функции **pp92_start**:

int **pp92_start** (Pp92 pp),

где:

– pp – объект, полученный ранее при выполнении функции pp92_open;

– возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.4 Функция «pp92_stop»

5.4.1 Для окончания обработки входящего трафика Предобработчику необходимо передать команду на окончание работы **pp92_stop**. Перед началом конфигурирования Предобработчика обработку трафика необходимо остановить. Остановка и начало обработки трафика сбрасывает накопленные статистические данные.

Синтаксис функции **pp92_stop**:

int **pp92_stop** (Pp92 pp),

где:

– pp – объект, полученный ранее при выполнении функции pp92_open;

– возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.5 Функция «pp92_is_running»

5.5.1 5.5.1 To get the state of the data processing on the Preprocessor, the function **p92_is_running** is used.

Синтаксис функции **pp92_is_running**:

int **pp92_is_running** (Pp92 pp),

где:

– Pp – объект, полученный ранее при выполнении функции pp92_open;

– возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает, что процесс обработки трафика запущен на Предобработчике.

5.6 Функция «pp92_get_ident»

5.6.1 Получение идентификационных данных Предобработчика выполняется функцией **pp92_get_ident.**

Синтаксис функции **pp92_get_ident**:

int **pp92_get_ident** (Pp92 pp, struct pp92_ident *ident),

где:

– pp – объект, полученный ранее при выполнении функции pp92_open;

– ident – структура данных типа pp92_ident, содержащая идентификационные данные Предобработчика;

– возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.7 Функция «pp92_get_health»

5.7.1 Получение данных состояния Предобработчика выполняется функцией **pp92_get_health.**

Синтаксис функции **pp92_get_health**:

int **pp92_get_health** (Pp92 pp, struct pp92_health *health),

где:

– pp – объект, полученный ранее при выполнении функции pp92_open;

– health – структура данных типа pp92_health, содержащая данные состояния Предобработчика;

– возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.8 Функция «pp92_set_flow_config»

5.8.1 Запись конфигурационных параметров Предобработчика выполняется функцией **pp92_set_flow_config.**

Синтаксис функции **pp92_set_flow_config**:

int **pp92_set_flow_config** (Pp92 pp, size_t flow, const struct pp92_flow_config *config)

где:

– pp – объект, полученный ранее при выполнении функции pp92_open;

– flow – индекс потока данных;

– config – структура данных типа pp92_flow_config, содержащая конфигурацию потока данных;

– возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.9 Функция «pp92_get_flow_config»

5.9.1 Чтение конфигурационных параметров Предобработчика выполняется функцией **pp92_get_flow_config.**

Синтаксис функции **pp92_get_flow_config**:

int **pp92_get_flow_config** (Pp92 pp, size_t flow, struct pp92_flow_config *config)

где:

- pp – объект, полученный ранее при выполнении функции pp92_open;
- flow – индекс потока данных;
- config – структура данных типа pp92_flow_config для получения конфигурации потока данных;
- возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.10 Функция «pp92_set_flow_filter»

5.10.1 Запись параметров фильтра потока данных Предобработчика выполняется функцией **pp92_set_flow_filter.**

Синтаксис функции **pp92_set_flow_filter**:

int **pp92_set_flow_filter** (Pp92 pp, size_t flow, const struct pp92_flow_filter *filter)

где:

- pp – объект, полученный ранее при выполнении функции pp92_open;
- flow – индекс потока данных;
- filter – структура данных типа pp92_flow_filter, содержащая конфигурацию фильтра потока данных;
- возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.11 Функция «pp92_get_flow_filter»

5.11.1 Чтение параметров фильтра потока данных Предобработчика выполняется функцией **pp92_get_flow_filter.**

Синтаксис функции **pp92_get_flow_filter**:

int **pp92_get_flow_filter** (Pp92 pp, size_t flow, struct pp92_flow_filter *filter)

где:

– pp – объект, полученный ранее при выполнении функции pp92_open;

– flow – индекс потока данных;

– filter – структура данных типа pp92_flow_filter для записи конфигурации фильтра потока данных;

– возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.12 Функция «pp92_get_flow_info»

5.12.1 Чтение параметров информации о потоке данных Предобработчика выполняется функцией **pp92_get_flow_info.**

Синтаксис функции **pp92_get_flow_info**:

int **pp92_get_flow_info** (Pp92 pp, size_t flow, struct pp92_flow_info *info)

где:

– pp – объект, полученный ранее при выполнении функции pp92_open;

– flow – индекс потока данных;

– info – структура данных типа pp92_flow_info для сохранения данных потока данных;

– возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.13 Функция «pp92_get_flow_stats»

5.13.1 Чтение данных статистики потока данных Предобработчика выполняется функцией **pp92_get_flow_stats.**

Синтаксис функции **pp92_get_flow_stats**:

int **pp92_get_flow_stats** (Pp92 pp, size_t flow, struct pp92_flow_stats *stats)

где:

– pp – объект, полученный ранее при выполнении функции pp92_open;

– flow – индекс потока данных;

– stats – структура данных типа pp92_flow_stats для сохранения данных статистики потока данных;

– возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.14 Функция «pp92_set_channel_config»

5.14.1 Запись конфигурационных данных измерительного канала выполняется функцией **pp92_set_channel_config.**

Синтаксис функции **pp92_set_channel_config**:

int **pp92_set_channel_config** (Pp92 pp, size_t channel, const struct pp92_channel_config *config)

где:

- pp – объект, полученный ранее при выполнении функции pp92_open;
- channel – индекс измерительного канала;
- config – структура данных типа pp92_channel_config, содержащая конфигурационные данные измерительного канала;
- возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.15 Функция «pp92_get_channel_config»

5.15.1 Чтение конфигурационных данных измерительного канала выполняется функцией **pp92_get_channel_config.**

Синтаксис функции **pp92_get_channel_config**:

int **pp92_get_channel_config** (Pp92 pp, size_t channel, struct pp92_channel_config *config)

где:

- pp – объект, полученный ранее при выполнении функции pp92_open;
- channel – индекс измерительного канала;
- config – структура данных типа pp92_channel_config для чтения конфигурационных данных измерительного канала;
- возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.16 Функция «pp92_get_channel_stats»

5.16.1 Чтение статистических данных измерительного канала выполняется функцией **pp92_get_channel_stats.**

Синтаксис функции **pp92_get_channel_stats**:

int **pp92_get_channel_stats** (Pp92 pp, size_t channel, struct pp92_channel_stats *stats)

где:

– pp – объект, полученный ранее при выполнении функции pp92_open;

– channel – индекс измерительного канала;

– stats – структура данных типа pp92_channel_stats для чтения статистических данных измерительного канала;

– возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.17 Функция «pp92_get_port_status»

5.17.1 Чтение данных статуса порта Ethernet выполняется функцией **pp92_get_port_status.**

Синтаксис функции **pp92_get_port_status**:

int **pp92_get_port_status** (Pp92 pp, size_t port, struct pp92_port_status *status)

где:

– pp – объект, полученный ранее при выполнении функции pp92_open;

– port – номер Ethernet порта;

– stats – структура данных типа pp92_port_status для чтения статуса по порту Ethernet;

– возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.18 Функция «pp92_get_port_status»

5.18.1 Чтение статистических данных порта Ethernet выполняется функцией **pp92_get_port_stats.**

Синтаксис функции **pp92_get_port_stats**:

int **pp92_get_port_stats** (Pp92 pp, size_t port, struct pp92_port_stats *stats)

где:

– pp – объект, полученный ранее при выполнении функции pp92_open;

– port – номер Ethernet порта;

– stats – структура данных типа pp92_port_stats для чтения статистических данных по порту Ethernet;

– возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.19 Функция «pp92_wait»

5.19.1 Функция **pp92_wait** обеспечивает запуск таймера на ожидание данных. Выход из данной функции осуществляется по истечении таймера или при получении нового блока данных.

Синтаксис функции **pp92_wait**:

int **pp92_wait** (Pp92 pp, int msec)

где:

- pp – объект, полученный ранее при выполнении функции pp92_open;
- msec – длина таймера в миллисекундах;
- возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.20 Функция «pp92_get_fd»

5.20.1 Функция **pp92_get_fd** позволяет получить идентификатор устройства в ОС Astra Linux. Данный идентификатор может быть использован для работы с Предобработчиком как с устройством ОС Astra Linux.

Синтаксис функции **pp92_get_fd**:

int **pp92_get_fd** (Pp92 pp92)

где:

- pp – объект, полученный ранее при выполнении функции pp92_open;
- возвращаемое функцией значение – идентификатор Предобработчика в ОС Linux или ошибка (значение -1).

5.21 Функция «pp92_dequeue»

5.21.1 Функция **pp92_dequeue** позволяет получить доступ к заполненному блоку данных в конце очереди. Полученный блок данных становится недоступным для Предобработчика. Данная функция обеспечивает чтение мгновенных значений и интегральных значений, полученных Предобработчиком.

Синтаксис функции **pp92_dequeue**:

int **pp92_dequeue** (Pp92 pp, struct pp92_block *block),

где:

- pp – объект, полученный ранее при выполнении функции pp92_open;

– block – структура данных типа pp92_block для доступа к мгновенным и интегральным значениям;

– возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.22 Функция «pp92_enqueue»

5.22.1 Функция **pp92_enqueue** позволяет освободить занятый функцией **pp92_dequeue** блок памяти для дальнейшего использования Предобработчиком.

Синтаксис функции **pp92_dequeue**:

int **pp92_enqueue** (Pp92 pp, const struct pp92_block *block),

где:

– pp – объект, полученный ранее при выполнении функции pp92_open;

– block – структура данных типа pp92_block, полученная ранее функцией **pp92_dequeue**;

– возвращаемое функцией значение – результат выполнения функции. Значение равное 0, означает успешное выполнение функции, значение отличное от 0, содержит код ошибки.

5.23 Функция «pp92_block_samples»

5.23.1 Функция **pp92_block_samples** используется для получения указателя на массив данных с мгновенными значениями из блока памяти, полученного функцией **pp92_dequeue.**

Синтаксис функции **pp92_block_samples**:

const void ***pp92_block_samples** (Pp92 pp, const struct pp92_block *block),

где:

– pp – объект, полученный ранее при выполнении функции pp92_open;

– block – структура данных типа pp92_block, полученная ранее функцией **pp92_dequeue**;

– возвращаемое функцией значение – ссылка на массив мгновенных значений.

5.24 Функция «pp92_block_aggregates»

5.24.1 Функция **pp92_block_aggregates** используется для получения указателя на массив данных с интегральными значениями из блока памяти, полученного функцией **pp92_dequeue.**

Синтаксис функции **pp92_dequeue**:

const void ***pp92_block_aggregates** (Pp92 pp, const struct pp92_block *block),

где:

– pp – объект, полученный ранее при выполнении функции pp92_open;

– block – структура данных типа pp92_block, полученная ранее функцией **pp92_dequeue**;

– возвращаемое функцией значение – ссылка на массив интегральных значений.

Массив интегральных значений содержит 5 параметров типа float.

С нулевым смещением хранится значением минимума (MIN).

Со смещением 1 для типа float хранится значение максимума (MAX).

Со смещением 2 для типа float хранится значение среднего арифметического (AVG).

Со смещением 3 для типа float хранится значение среднего квадратичного (RMS).

Пример использования функции **pp92_block_aggregates**:

```
/* Index of MIN in aggregate parameters */
#define PP92_AGGREGATE_MIN        0
/* Index of MAX in aggregate parameters */
#define PP92_AGGREGATE_MAX        1
/* Index of AVG in aggregate parameters */
#define PP92_AGGREGATE_AVG        2
/* Index of RMS in aggregate parameters */
#define PP92_AGGREGATE_RMS        3
/* Count of aggregate parameters */
#define PP92_AGGREGATES_COUNT    4
```

```
const float *aggregates = pp92_block_aggregates(pp92, block);
char line[80];
snprintf(line, sizeof(line), "CH%"PRIu8": min=%.2f, max=%.2f, avg=%.2f, rms=%.2f\r\n",
        block->channel,
        aggregates[PP92_AGGREGATE_MIN],
        aggregates[PP92_AGGREGATE_MAX],
        aggregates[PP92_AGGREGATE_AVG],
        aggregates[PP92_AGGREGATE_RMS]
);
```

Приложение А – Характеристики профиля «HVDC» потока 9-2

А.1 В настоящем приложении приведено описание спецификации профиля «HVDC» потока 9-2, поддерживаемого (распознаваемого) устройством сопряжения.

А.2 Спецификация профиля «HVDC» потока 9-2, принимаемого и распознаваемого устройством сопряжения, в целом соответствует спецификации потока 9-2 по стандарту IEC 61850-9-2 (версий 1.0/2.0 указанного стандарта) с ограничениями и отличиями, описанными далее в пп. А.3 и А.4.

А.3 В стандартной спецификации IEC 61850-9-2 размер поля SmpCnt в APDU Ethernet-фреймов потока 9-2 ограничен 2-мя байтами (тип значения – INT16U), что ограничивает частоту дискретизации передаваемого потока 9-2 величиной не более 65536 Гц.

Для обеспечения же передачи потока 9-2 типа «HVDC» с повышенной частотой дискретизации 100 000 Гц размер значения поля SmpCnt в составе APDU Ethernet-фреймов потока 9-2 типа «HVDC» расширен до 4 байт (приведен к типу INT32U).

При этом регламент изменения значения поля SmpCnt в APDU Ethernet-фреймов потока 9-2 типа «HVDC» сохранен в соответствии со спецификацией IEC 61850-9-2, в частности:

- значение SmpCnt увеличивается на 1 с каждым следующим отсчетом мгновенных значений;
- значение SmpCnt устанавливается в 0 (обнуляется) для мгновенного значения, момент времени среза (аналогового сигнала) которого совпадает с границей секунд точного астрономического времени.

Примечание: Таким образом, значение SmpCnt в Ethernet-фреймах потока 9-2, соответствующего профилю «HVDC», принимает последовательные значения в диапазоне от 0 до 99 999.

А.4 В одном отсчёте мгновенных значений в Ethernet-фрейме потока 9-2, соответствующего профилю «HVDC», (т.е. в блоке данных ASDUEthernet-фрейма указанного потока), может передаваться только одно мгновенное значение электрической величины (тип величины – «напряжение»). Указанное мгновенное значение, передаваемое в блоке данных ASDUEthernet-фрейма потока, должно иметь тип SAV («sampledvalue») в соотв. со спецификацией IEC 61850-7-3 и содержать в себе только два атрибута данных:

- “instMag” (тип значения – “AnalogueValue”) – содержит в себе только целочисленный вариант значения (“i”) (тип значения –

INT32); при этом предполагается, что sVC.scaleFactor = 0,01, sVC.offset = 0 (см. ниже примечание);

– “q” (тип “Quality”), закодированный в соответствии со спецификацией IEC 61850-8-1 (см. пункт 8.2 спецификации IEC 61850-8-1) в виде 32-битной строки.

Примечание: Сам атрибут данных “sVC” (с полями “scaleFactor и “offset”) не должен передаваться дополнительным атрибутом в мгновенном значении (типа SAV) в ASDU Ethernet-фрейма потока 9-2, соответствующего профилю «HVDC». В связи с этим, значения “scaleFactor и “offset” должны приниматься, как указано выше (т.е. scaleFactor = 0,01, offset = 0). При этом фактическая (целевая) величина мгновенного значения измеряемого напряжения в вольтах должна вычисляется (на основании значения, передаваемого в атрибуте данных “instMag”) по формуле:

$$U_1^{inst} = (instMag\ .i) \times scaleFacto\,r + offset\ .$$

Приложение Б. Нормативные ссылки

Обозначение (номер) документа	Наименование документа
IEC 61850-9-2	Communication networks and systems for power utility automation – Part 9-2: Specific communication service mapping (SCSM) – Sampled values over ISO/IEC 8802-3
IEC 61850-9-2 LE	Implementation Guideline for Digital Interface to Instrument Transformers using IEC 61850-9-2 (UCA International Users Group, 05-08-2005)