ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ
НАУЧНО-ПРОИЗВОДСТВЕННОЕ ПРЕДПРИЯТИЕ «ЭКРА»

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ ЭКРА.00140-01 «G119_GCC ГЕНЕРАТОР КОНТРОЛЬНОГО ТОКА Г119»

Описание программы ЭКРА.00140-0113

2025

Авторские права на данную документацию принадлежат ООО НПП «ЭКРА» (г. Чебоксары). Снятие копий или перепечатка разрешаются только по согласованию с разработчиком.

АННОТАЦИЯ

Настоящий документ является описанием программного обеспечения ЭКРА.00140-01 13 «G119_GCC генератор контрольного тока Г119» (далее – ПО или «G119_GCC»).

В документе приведены сведения применения, структуры и функционирования программы «G119_GCC». Также рассматриваются вопросы о входных и выходных данных, используемых в системе, конфигурации технических средств, необходимых для функционирования программы.

Содержание

1. Общие сведения 6
1.1. Обозначение и наименование программы 6
1.2. Программное обеспечение, необходимое для функционирования программы 6
1.3. Языки программирования, на которых написана программа 6
2. Функциональное назначение 7
2.1. Регистрация изменений в журнал событий 7
2.2. Приём входных, формирование выходных дискретных сигналов и управление выходом инвертора 7
2.3. Инициализация модуля времени 7
2.4. Связь с внешними устройствами через цифровой интерфейс 8
2.5. Инициализация модуля индикации 8
2.6. Самодиагностика 8
2.7. Инициализация модуля измерений 8
3. Описание логической структуры 9
3.1. Общая структура ПО 9
3.1.1. Дистрибутив ПО 9
3.1.2. Логическая структура ПО 9
3.2. Алгоритм компонентов ПО 9
3.2.1. Программа загрузчика 9
3.2.2. Программа приложения 10
4. Связь со сторонним ПО 28
5. Используемые технические средства 29
6. Вызов и загрузка 30
7. Входные и выходные данные 31
7.1. Входные данные 31
7.2. Выходные данные 31

Обозначения и сокращения

ГКТ	генератор контрольного тока;
ПО	программное обеспечение;
ОЗЗ	однофазное замыкание на землю;
ШИМ	широтно-импульсная модуляция;
COM	communications port (последовательный порт);
EEPROM	electrically erasable programmable read-only memory (электрически стираемое перепрограммируемое ПЗУ, один из видов энергонезависимой памяти);
Modbus/RTU	открытый коммуникационный протокол для организации связи между электронными устройствами по линии связи RS485;
RTU	remote terminal unit (устройство связи с объектом);

1. Общие сведения

1.1. Обозначение и наименование программы

Наименование программного обеспечения: «G119_GCC генератор контрольного тока Г119».

Краткое наименование программы: «G119_GCC».

1.2. Программное обеспечение, необходимое для функционирования программы

«G119_GCC» поставляется заказчикам в памяти Г119, а в случае обновления ранее установленной версии - на электронном носителе информации или по электронной почте. Для работы программы не требуется никакого дополнительного программного обеспечения.

1.3. Языки программирования, на которых написана программа

Компоненты «G119_GCC» разработаны на следующих языках программирования: C++, C и Ассемблер.

2. Функциональное назначение

«G119_GCC» предназначено для контроля степени расстройки компенсации емкостного тока и обеспечения резонансной настройки дугогасящих реакторов.

Программа написана с использованием операционной системы реального времени FreeRTOS. «G119_GCC» должно обеспечивать следующие основные функции:

- регистрация изменений в журнал событий;
- прием входных, формирование выходных дискретных сигналов и управление выходом инвертора;
- инициализация модуля времени;
- связь с внешними устройствами через цифровой интерфейс;
- инициализация модуля индикации;
- самодиагностика;
- инициализация модуля измерений.

2.1. Регистрация изменений в журнал событий

Журнал событий должен хранится во внешней энергозависимой памяти EEPROM. Журнал событий должен обеспечивать выполнение следующих функций:

- создавать очередь записи в журнал событий;
- чтение событий из журнала событий;

2.2. Приём входных, формирование выходных дискретных сигналов и управление выходом инвертора

Дискретные входа должны опрашиваться каждые 250 микросекунд в прерывании. «G119_GCC» должно обеспечивать:

- чтение дискретных сигналов;
- проверку наличия напряжения;
- управление выходом инвертора согласно выбранному режиму;
- управление дискретными выходами;
- слежение за параметрами системы.

2.3. Инициализация модуля времени

Модуль времени должен обеспечивать выполнение следующих функций:

- синхронизацию времени с внутренним RTC;
- фиксация времени от начала работы устройства;
- обеспечение временными задержками без блокировки задачи.

2.4. Связь с внешними устройствами через цифровой интерфейс

Модуль связи должен обеспечивать выполнение следующих функций:

- инициализация регистров Modbus;
- создание обработчиков периферии USART1, USART3;
- создание задач обработки пакетов ModbusRTU для каждой периферии.

«G119_GCC» должно обеспечивать возможность программной конфигурации функций с помощью специального комплекса программ и обмен данными с внешними устройствами по следующим протоколам:

- Modbus RTU.

2.5. Инициализация модуля индикации

Модуль индикации должен обеспечивать выполнение следующих функций:

- сообщать о состоянии параметров в системе;
- сообщать о состоянии температуры.

2.6. Самодиагностика

«G119_GCC» должно обеспечивать:

- непрерывную диагностику самой себя и аппаратных компонентов;
- сигнализацию об обнаруженных неисправностях;
- контроль работоспособности каналов связи.

2.7. Инициализация модуля измерений

«G119_GCC» должно обеспечивать:

- измерение частоты и RMS сети синхронизации;
- измерение RMS тока выхода инвертора;
- измерение температуры;
- измерение питания моста 48В;
- измерение напряжения открытия ключей.

3. Описание логической структуры

3.1. Общая структура ПО

3.1.1. Дистрибутив ПО

ПО «G119_GCC» состоит из компонентов загрузчика и приложения.

Файлы компонентов программ загрузчика и приложения. Соответствие между файлами программ представлено в таблице 3.

Таблица 1 - Соответствие между файлами программ

Наименование файла	Тип блока
G119_Boot_v1_0_0_0.hex	Прошивка программы загрузчика
G119_v1_0_0_xxx.hex	Прошивка программы приложения
* Символ «xxx» заменяет любое количество цифр, букв и специальных символов в обозначении названия программы.	

3.1.2. Логическая структура ПО

ПО «G119_GCC» выполняет следующие основные функции:

- регистрация изменений в журнал событий;
- прием входных и формирование выходных дискретных сигналов, управление выходом инвертора;
- инициализация модуля времени;
- связь с внешними устройствами через цифровой интерфейс;
- инициализация модуля индикации;
- самодиагностика;
- инициализация модуля измерений.
- ПО «G119_GCC» состоит из следующих программных модулей по функциональному назначению:
- прием входных дискретных сигналов;
- управление выходами инвертора согласно выбранному режиму;
- связь и обеспечение человеко-машинного интерфейса.

3.2. Алгоритм компонентов ПО

3.2.1. Программа загрузчика

Программа загрузчика выполняет задачу записи основной программы на устройство.

Программа осуществляет:

- инициализацию периферии: настройка системного таймера (Systick), тактирования (Clock) и аппаратного уровня (HAL).
- проверку условий запуска: анализ специального запроса на необходимость остаться в загрузчике.

– проверку целостности основной программы: проверка, не повреждена ли главная прошивка устройства.

3.2.2. Программа приложения

Программа написана с использованием операционной системы реального времени FreeRTOS. «G119_GCC» осуществляет следующие основные функции:

- журнал событий;
- дискретные входа;
- модуль времени;
- модуль связи;
- модуль индикации;
- самодиагностика;
- модуль управления выходом инвертора;
- модуль измерений.

3.2.2.1. Журнал событий

Журнал событий хранится во внешней энергонезависимой памяти EEPROM. Журнал может хранить до 10000 записей о событиях.

3.2.2.1.1 Очередь записи в журнал событий

Программа организует запись событий в журнал с помощью механизма отложенной записи. При вызове функций, таких как journal_events::WriteCode(…), данные не записываются напрямую в энергонезависимую память, а помещаются в очередь, расположенную в области SRAM. Эта область памяти размещена в секции .AppNoInit, которая не переинициализируется при перезапуске загрузчика или основного приложения. Такой подход обеспечивает сохранение событий в очереди даже в случае возникновения исключительных ситуаций, например, HardFault, и позволяет выполнить их запись позже. Кроме того, использование очереди минимизирует время блокировки других задач, что положительно сказывается на общей производительности системы.

Таблица 2 – Структура очереди для записи в EEPROM (структура JournalEvent_Pdu)

Наименование	Размер	Значение
Ключ инициализации: 1) Ключ присутствия EEPROM 2) Ключ для запроса на сохранение настроек	8 байт	1) 0xAC0000CA 2) 0x00BBAA00
Код события	2 байт	
Время события в мс	8 байт	
Данные записи [kJournalEventsDataSize]	kJournalEventsDataSize байт	

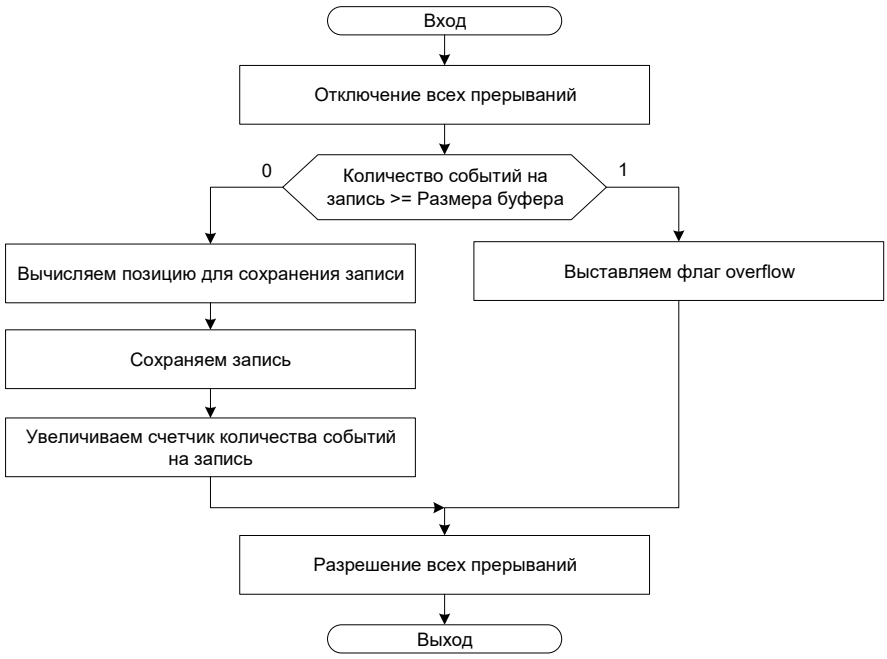


Рисунок 1 – Алгоритм выполнения сохранения данных события в буфер

3.2.2.1.2. Чтение событий из журнала

Чтение событий журнала производится непосредственно из EEPROM. Чтение используется в модуле связи, в котором ограничено количество записей для одновременного чтения до пяти (*REG_J_EVENTS_READ_COUNT* в registers.hpp). Тем самым это также уменьшает задержку процесса обработки пакета по каналу связи.

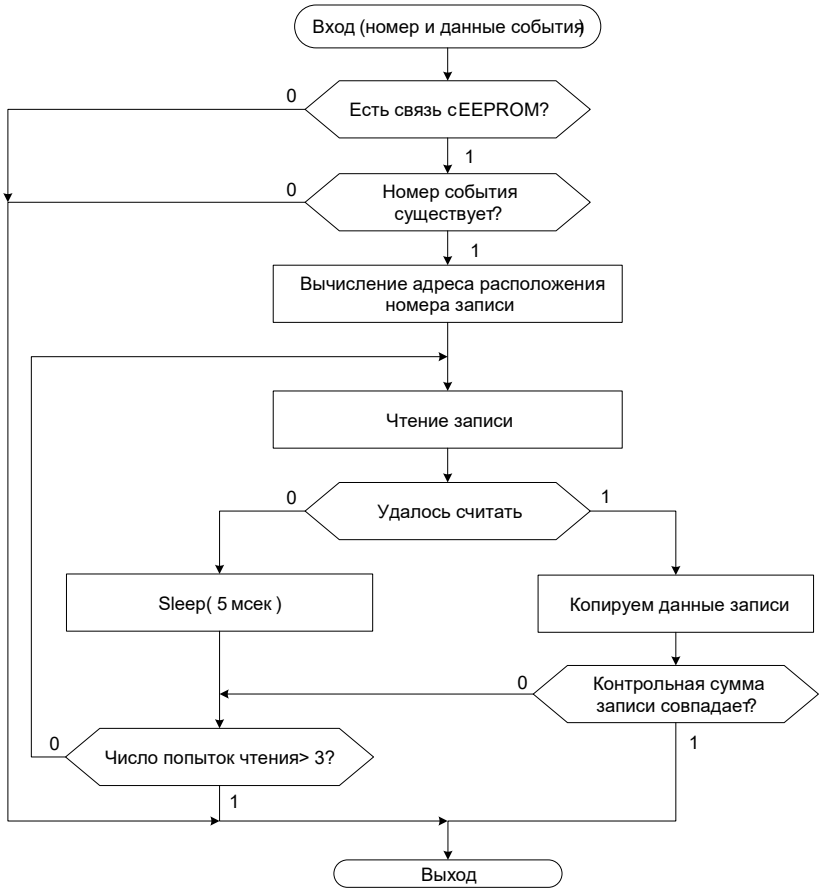


Рисунок 2 – Алгоритм чтения записи из журнала событий journal_events::*ReadEvent(…)*

3.2.2.1.3. Формат записей, хранящихся в EEPROM

Таблица 3 – Формат записей журнала событий (структура JournalEvent)

Тип данных	Размер	Значение
Преамбула	1 байт	0x68
Контрольная сумма (crc8)	1 байт	
Номер записи	2 байта	0..65535
Размер данных	2 байта	0..65535
Данные события	8 байт	

Записи в EEPROM выравнены по 32 байта, хотя сама запись занимает меньше. Оставшиеся в резерве 32 байта выбраны для более оптимального поиска последней записи.

3.2.2.1.4. Алгоритм задачи обслуживания журнала

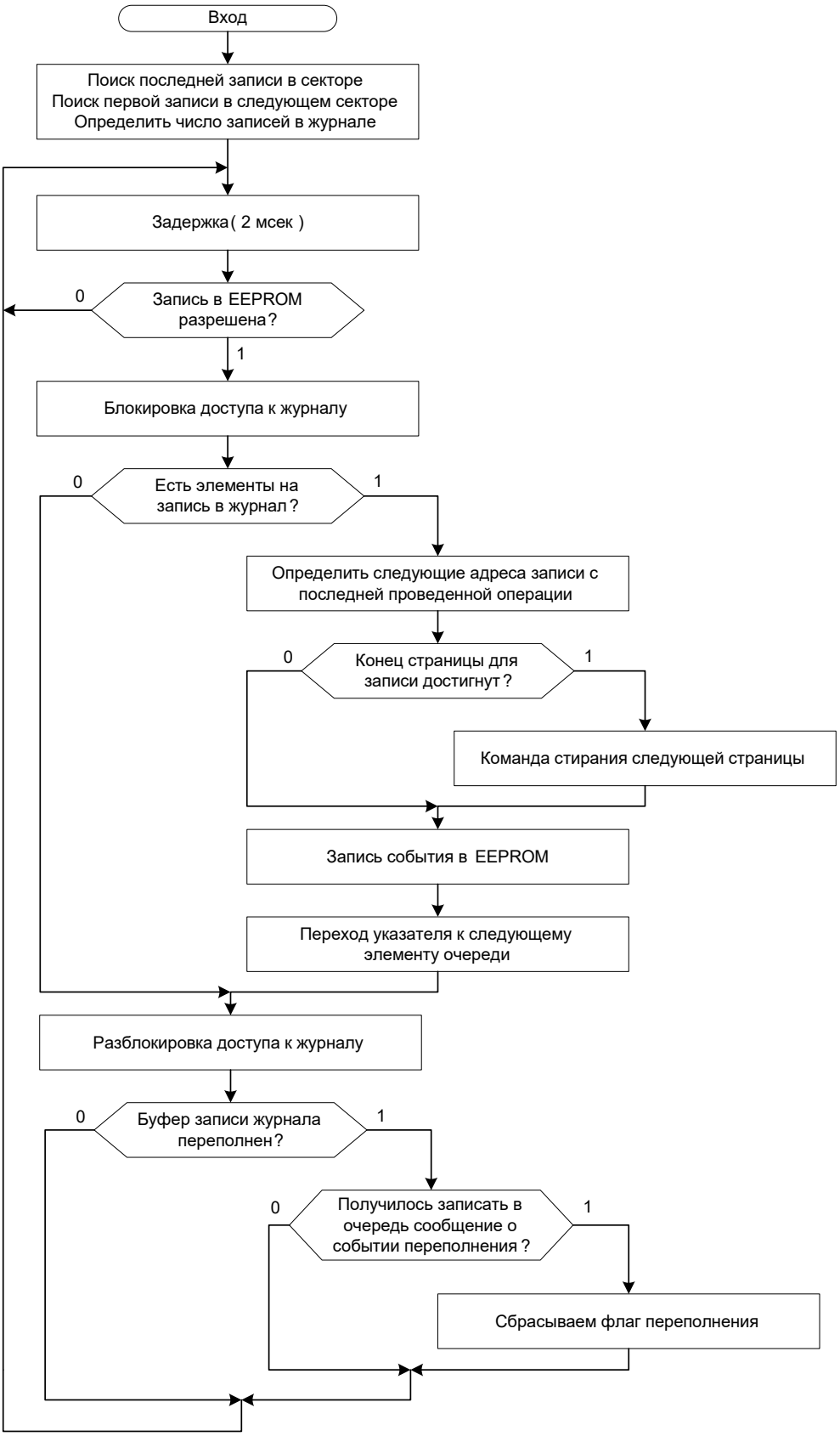


Рисунок 3 – Алгоритм задачи обслуживания журнала событий *JournalEvents_Task()*

3.2.2.2. Дискретные входа

Дискретные входа опрашиваются каждые 250 микросекунд в прерывании таймера (*TIM4_IRQHandler*).

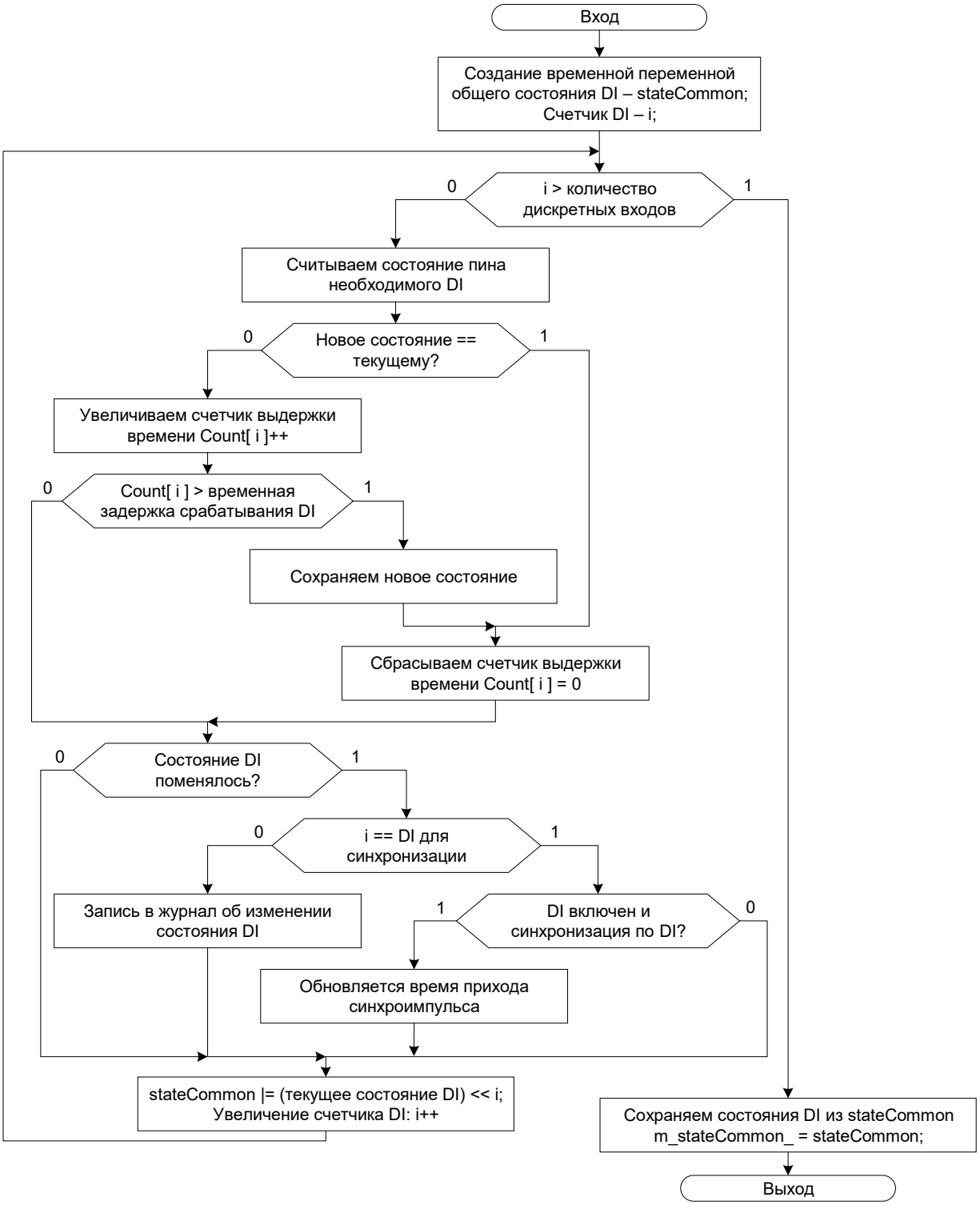


Рисунок 4 – Алгоритм опроса дискретных входов *discrete_input::Work()*

3.2.2.3. Модуль времени

Модуль времени предоставляет:

– время в формате UTC:

  – Time() – UTC время в секундах;

  – Time_ms() - UTC время в миллисекундах;

  – Time_mks() - UTC время в микросекундах;

– внутреннее время от начала работы устройства (непрерывное время):

  – Clock_ms() – внутреннее время в миллисекундах; □ Clock_mks() – внутреннее время в микросекундах; – временные задержки без блокировки задачи:

  – Delay(ms) – задержка в миллисекундах; □ Delay_mks(mks) - задержка в микросекундах; – установку текущего времени UTC.

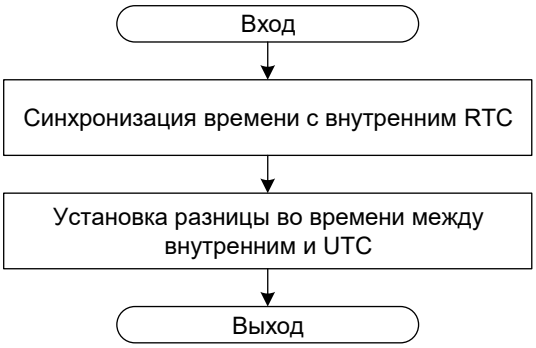


Рисунок 5 – Алгоритм инициализации времени *time::Init()*

3.2.2.4. Модуль связи

Алгоритм работы модуля связи работает таким образом, что сначала идет подготовка области памяти (регистры) для хранения данных, к которым будет осуществляться доступ по протоколу Modbus. Настраивает аппаратные последовательные порты (UART), которые будут использоваться для физической передачи данных по протоколу Modbus RTU. Настраивает аппаратные последовательные порты (UART), которые будут использоваться для физической передачи данных по протоколу Modbus RTU.

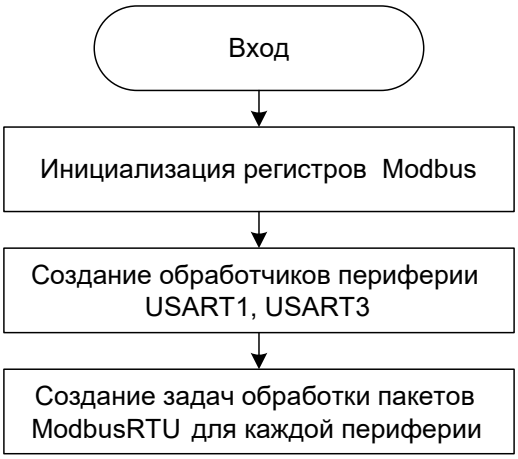


Рисунок 6 – Алгоритм главного модуля связи приложения *link::Init()*

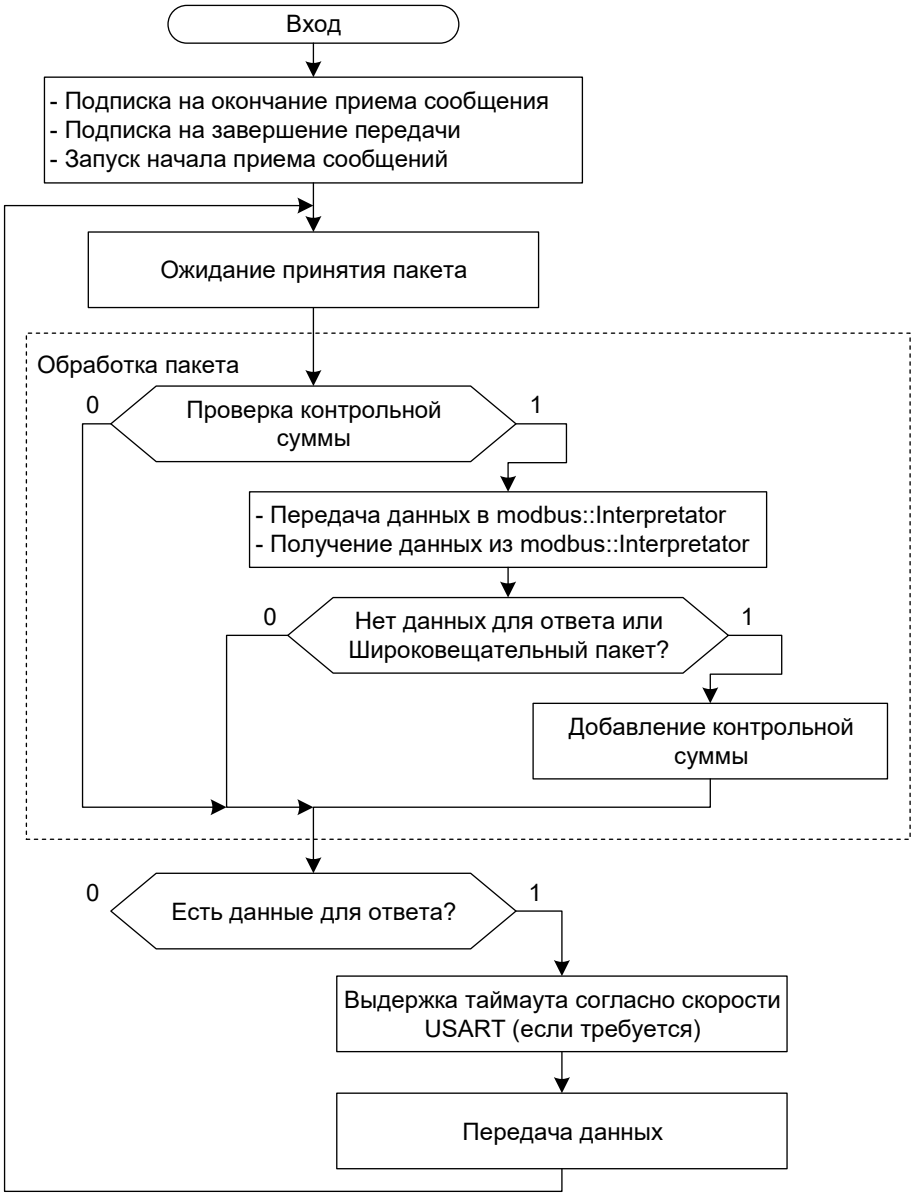


Рисунок 7 – Алгоритм задачи Modbus RTU приложения *Slave_Task()*

Метод *modbus::Interpretator()* интерпретирует пакет по кодам функций, осуществляет обновление регистров или запись данных с регистров согласно коду функции. Доступные коды функций и таблица регистров представлены в Приложении Б.

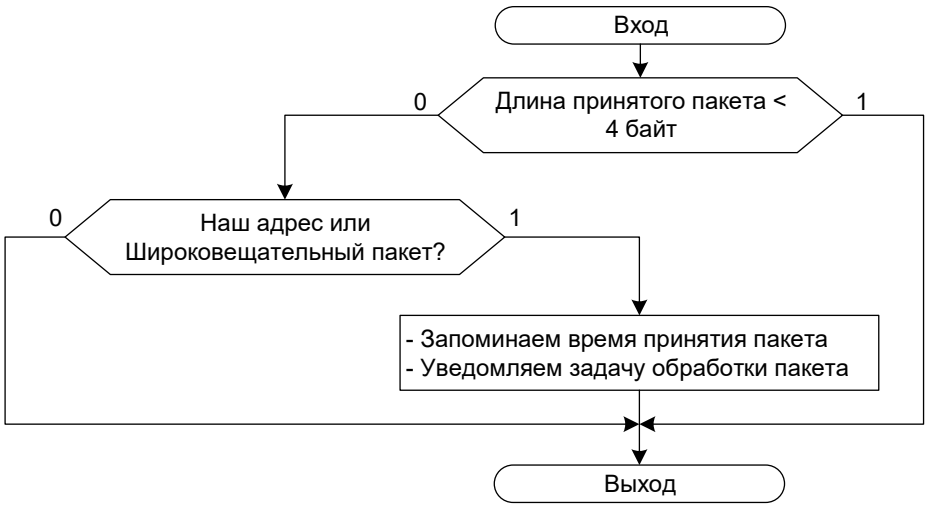


Рисунок 8 – Обработка прерывания по окончанию приема сообщения RS485 приложения *link::NotifyReceived_SlaveRsx_FromIrq()*

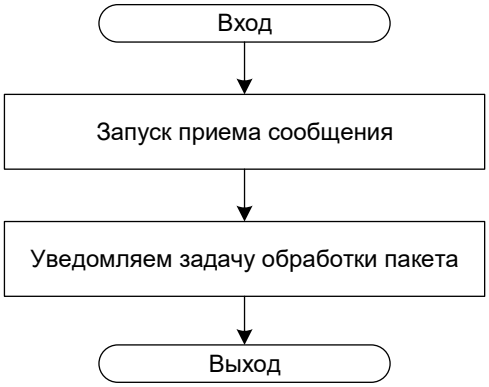


Рисунок 9 – Обработка пакета по окончанию передачи сообщения RS485 приложения *link::NotifyTransmited_SlaveRsx_FromIrq()*

3.2.2.5. Модуль индикации

Основная задача: управление светодиодами согласно режимам работы, наличию ошибок.

В режиме «Ошибка/Авария» зеленый светодиод гаснет, а красный включается на постоянное свечение.

В режиме «Сигнализация» реализовано мигание: оба светодиода одновременно загораются на короткое время, затем следует более длительная пауза с выключенными индикаторами.

В «Нормальном состоянии» устройство циклически подает короткие зеленые импульсы: зеленый светодиод кратковременно загорается, после чего следует длительная пауза, а красный светодиод всегда выключен.

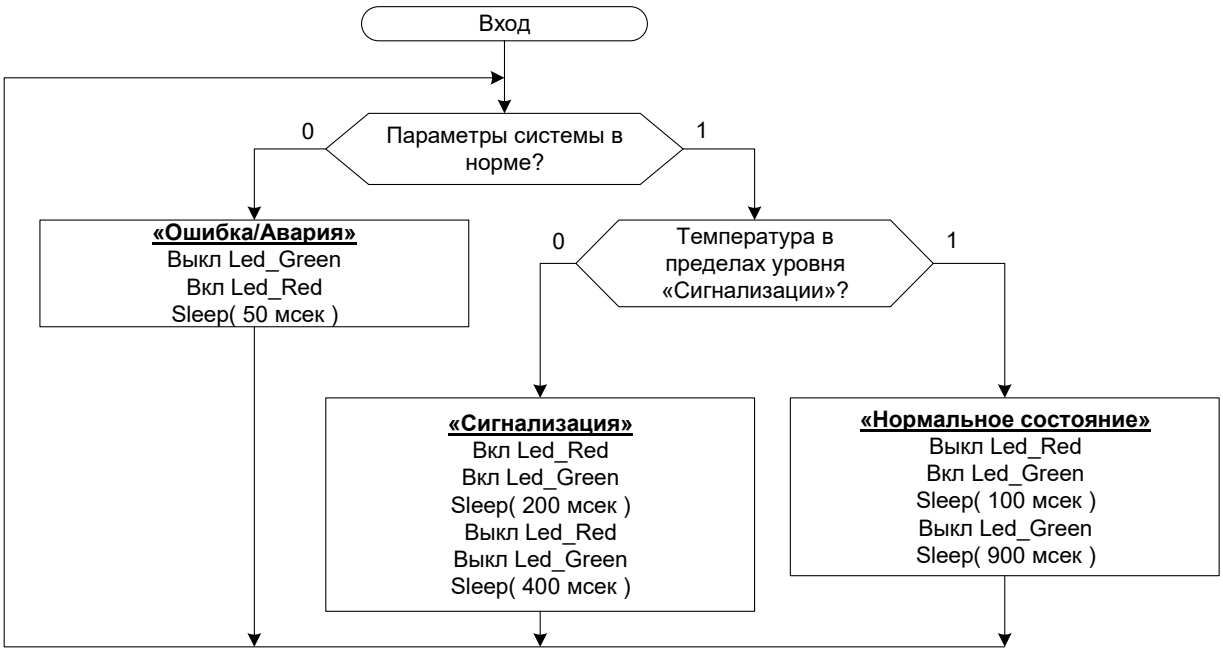


Рисунок 10 – Алгоритм задачи индикации *Indication_Task()*

3.2.2.6. Самодиагностика

Самодиагностика обеспечивает непрерывную диагностику самой себя и аппаратных компонентов. Включает сигнализацию об обнаруженных неисправностях. Обеспечивает контроль работоспособности каналов связи.

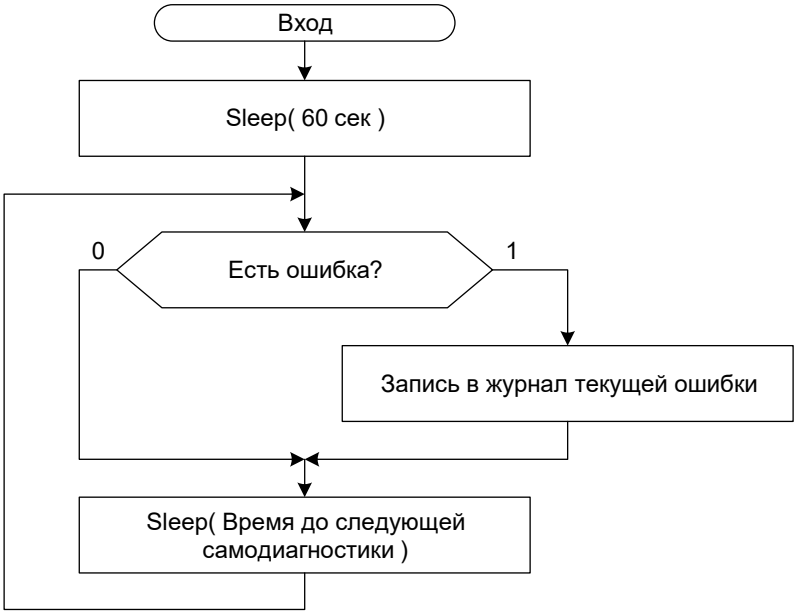


Рисунок 11 – Алгоритм задачи самодиагностики *SelfTest_Task()*

3.2.2.7. Инициализация модуля управления выходом инвертора

Управление выходом инвертора включает в себя 2 основных модуля:

1) Логика управления *«logic.cpp»*;

2) Управление мостом *«bridge.cpp»*.

3.2.2.7.1. Логика управления выходом инвертора Основная задача:

– управление выходом инвертора согласно выбранному режиму;
– управление дискретными выходами;
– слежение за параметрами системы;

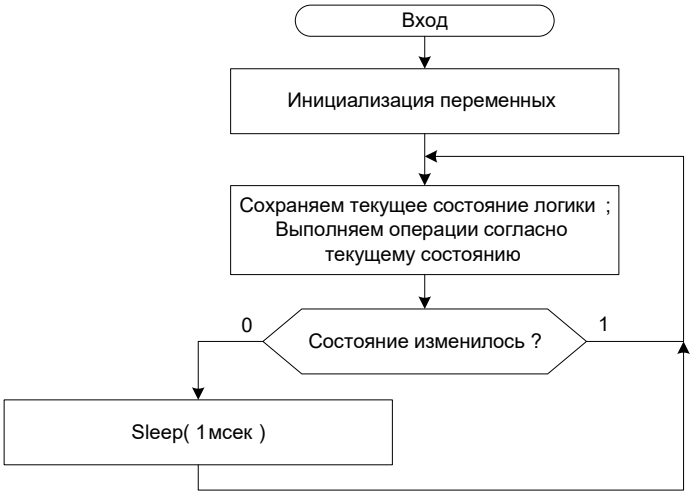
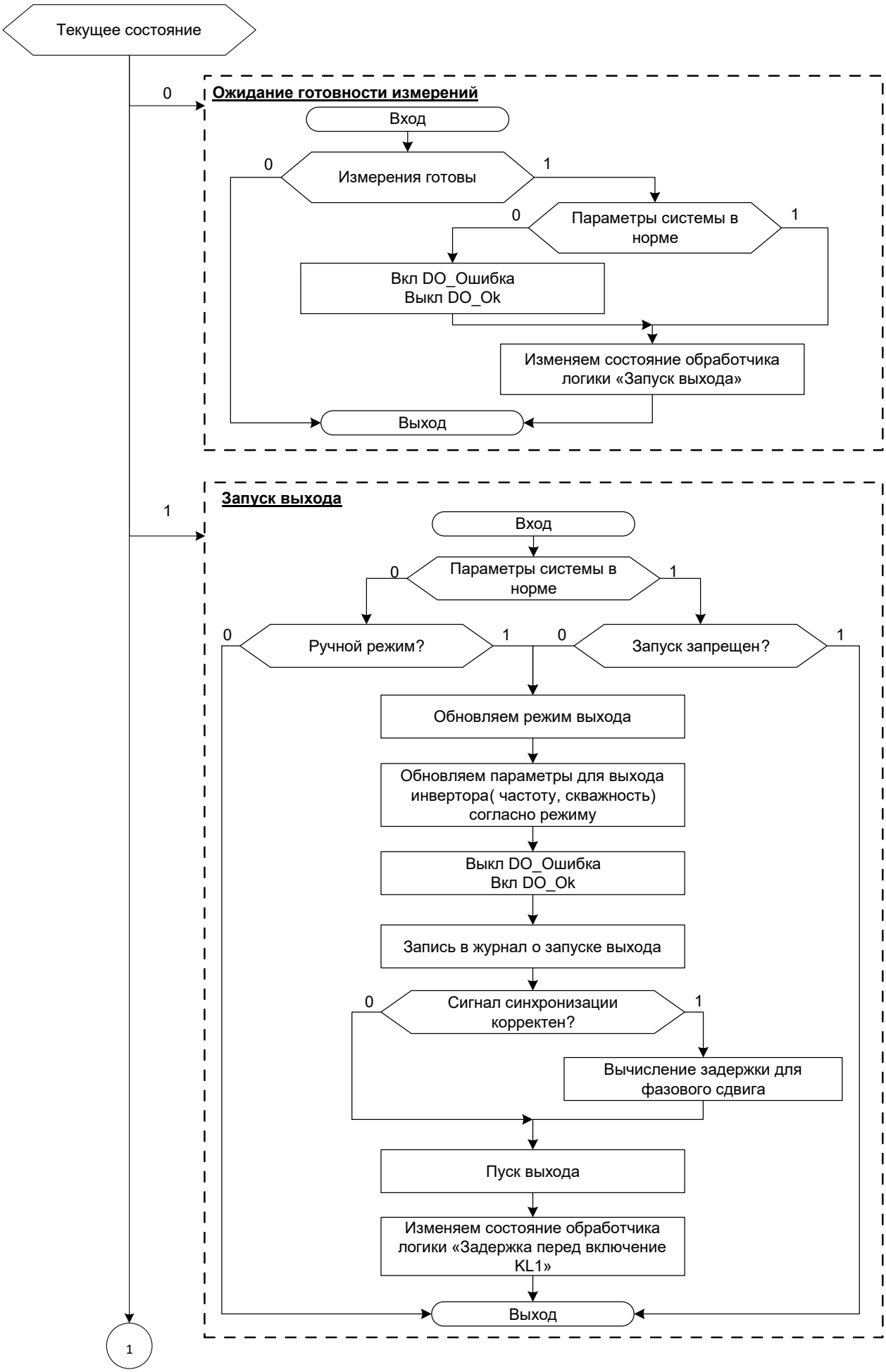
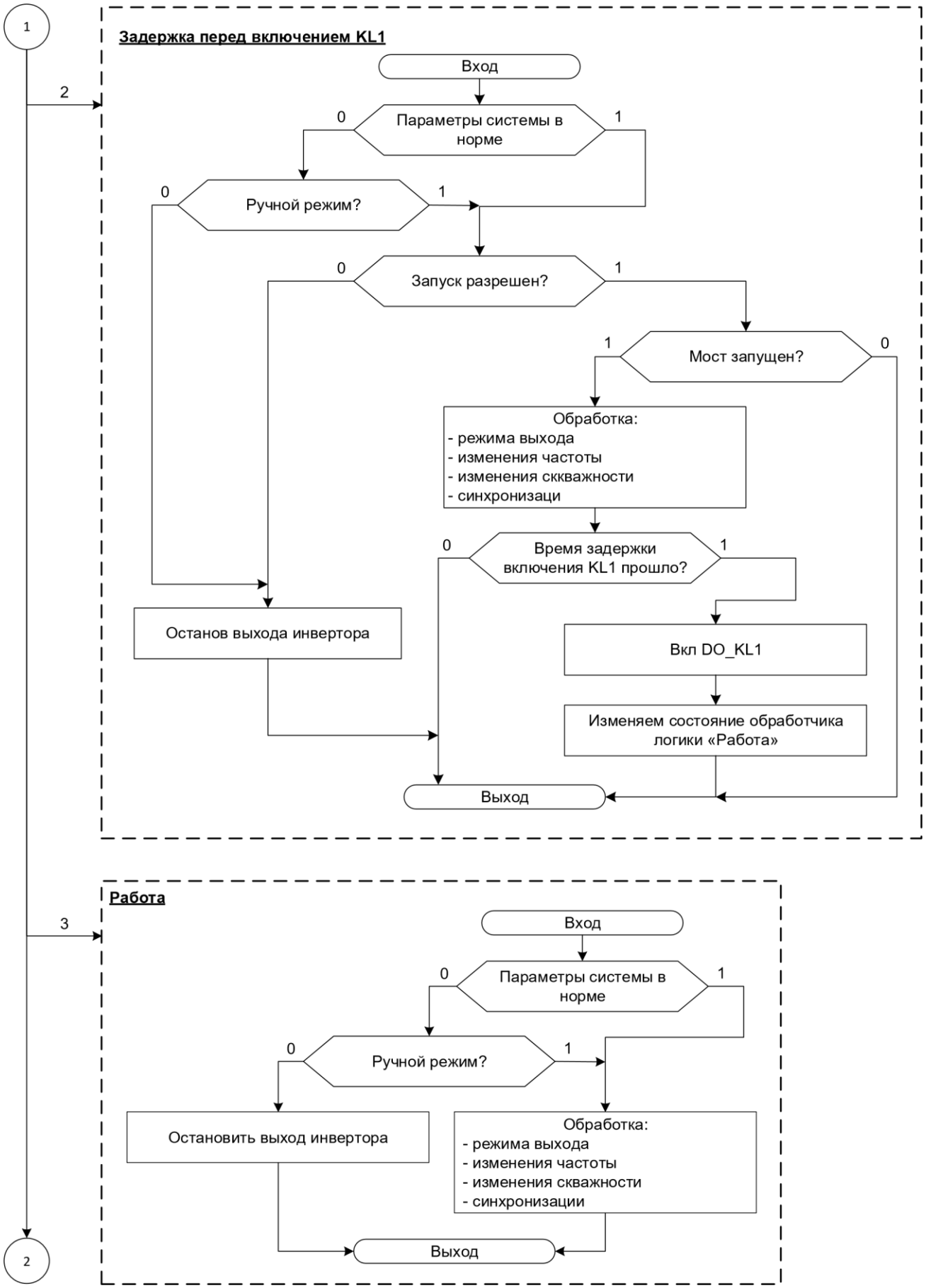


Рисунок 12 – Алгоритм задачи управления логикой выхода инвертора *Logic_Task()*


Текущее состояние
0
Ожидание готовности измерений
Вход
0
Измерения готовы
1
0
Параметры системы в норме
1
Вкл DO_Ошибка
Выкл DO_Ok
Изменяем состояние обработчика логики «Запуск выхода»
Выход
1
Запуск выхода
Вход
0
Параметры системы в норме
1
0
Ручной режим?
1
0
Запуск запрещен?
1
Обновляем режим выхода
Обновляем параметры для выхода инвертора(частоту, скважность) согласно режиму
Выкл DO_Ошибка
Вкл DO_Ok
Запись в журнал о запуске выхода
0
Сигнал синхронизации корректен?
1
Вычисление задержки для фазового сдвига
Пуск выхода
Изменяем состояние обработчика логики «Задержка перед включение KL1»
Выход
1


1
2
Задержка перед включением KL1
Вход
Параметры системы в норме
0
1
Ручной режим?
0
1
Запуск разрешен?
0
1
Мост запущен?
1
0
Обработка:
- режима выхода
- изменения частоты
- изменения сккважности
- синхронизаци
Время задержки включения KL1 прошло?
0
1
Останов выхода инвертора
Вкл DO_KL1
Изменяем состояние обработчика логики «Работа»
Выход
3
Работа
Вход
Параметры системы в норме
0
1
Ручной режим?
0
1
Остановить выход инвертора
Обработка:
- режима выхода
- изменения частоты
- изменения скважности
- синхронизации
Выход
2

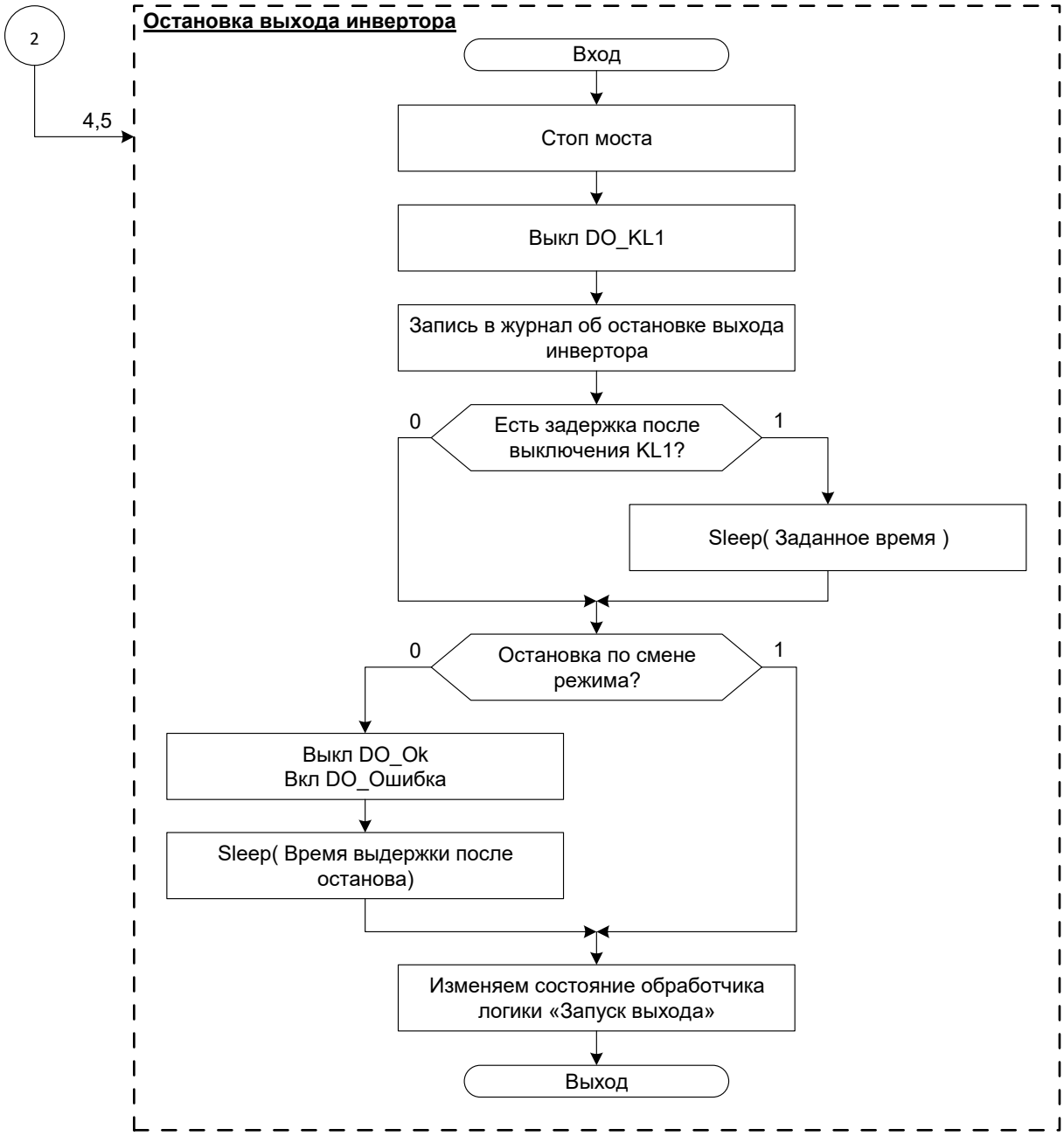


Рисунок 13 – Алгоритм выполнения операции согласно текущему режиму работы

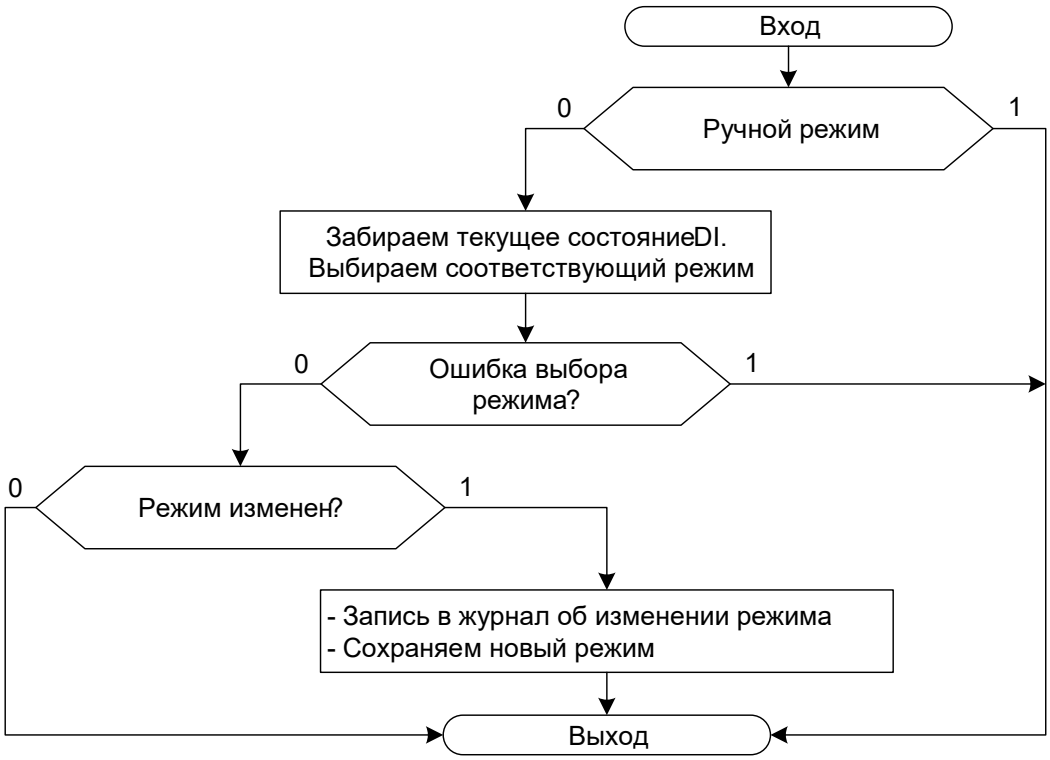


Рисунок 14 – Алгоритм обработки смены режима выхода *workMode()*

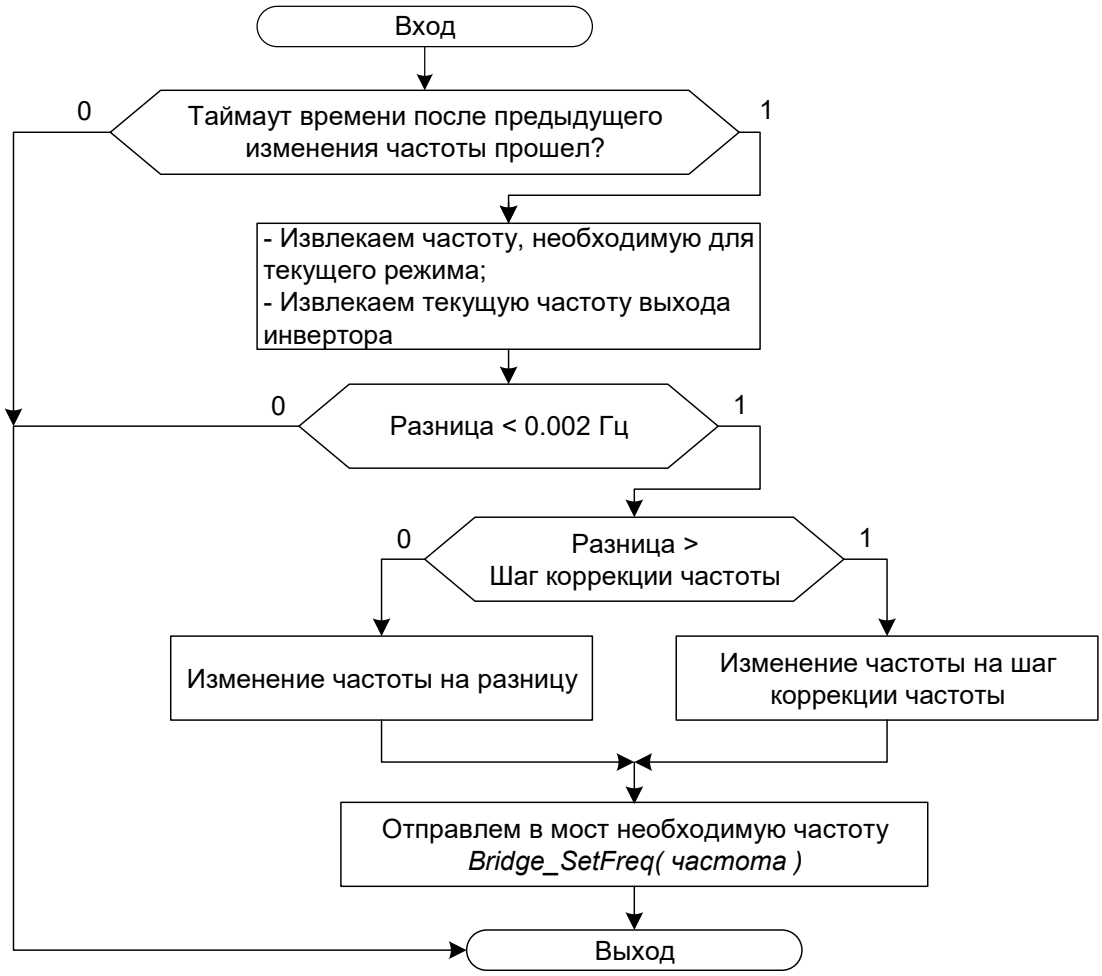


Рисунок 15 – Алгоритм обработки изменения частоты *workFreq()*

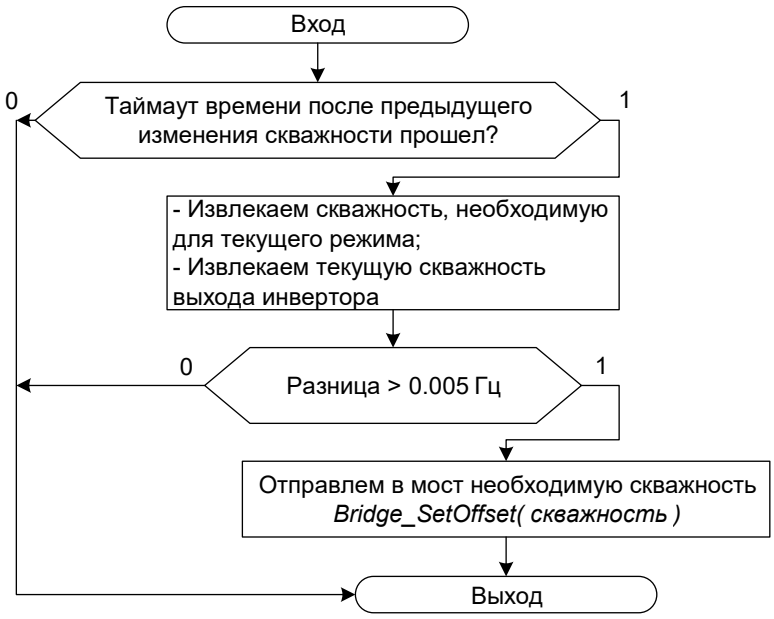


Рисунок 16 – Алгоритм обработки изменения скважности *workFill()*

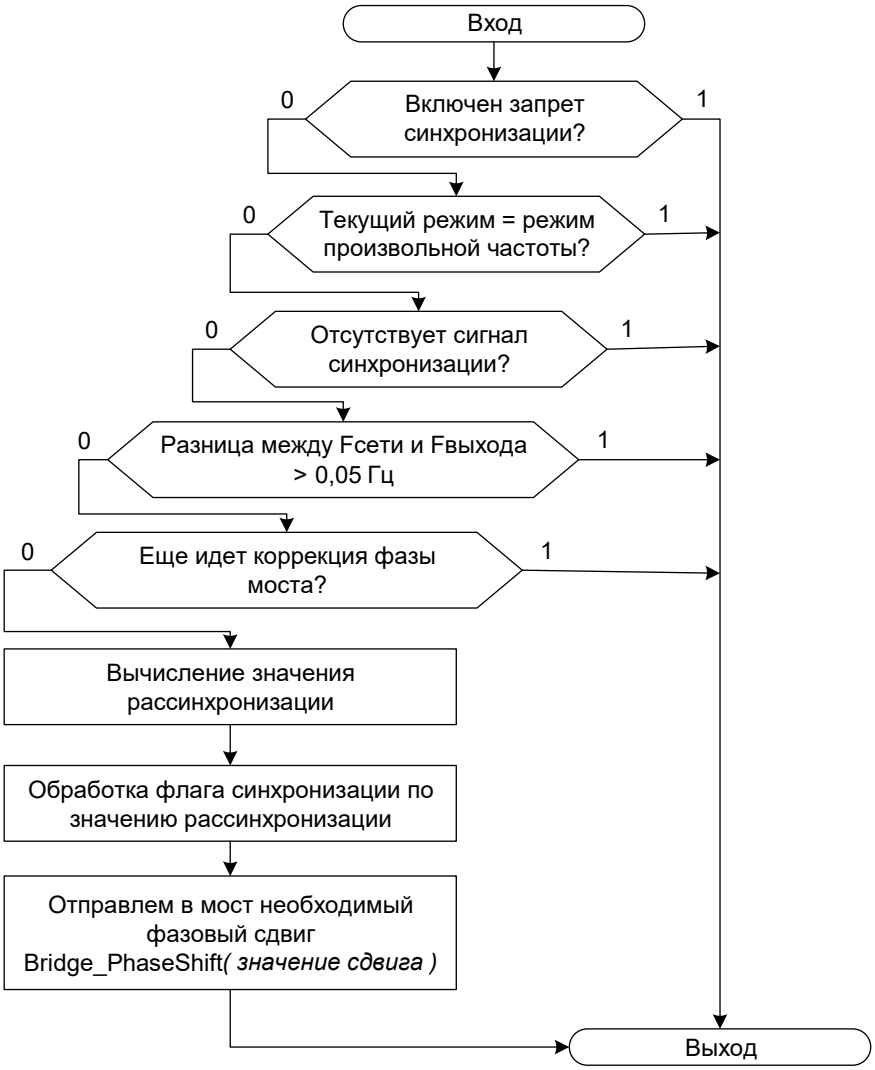


Рисунок 17 – Алгоритм выполнения синхронизация *workSynchro()*

3.2.2.7.2. Управление мостом

Управление плечами моста выполняет сам процессор переключением пинов, а не периферия. Для определения времени, когда нужно переключить плечи, задействован TIM8. Для задания времени прерывания служат функции класса времени:

– *SetPeriod( )* – устанавливает период таймера; – *Set_CCx( ):*

- *Set_CCx(1)* – устанавливает время Закрытия плеча 1 и Открытия плеча 2;
- *Set_CCx(2)* – устанавливает время Закрытия плеча 4 и Открытия плеча 3;
- *Set_CCx(3)* – устанавливает время Закрытия плеча 3 и Открытия плеча 4;

Закрытия плеча 2 и Открытия плеча 1 происходит в начале периода таймера.

Важно! Само изменения частоты, скважности, сдвига выполняется не в момент вызова соответствующих функций *SetFrequency(),SetOffset(),PhaseShiftWok(),* а обрабатывается в начале периода таймера и в прерываниях.

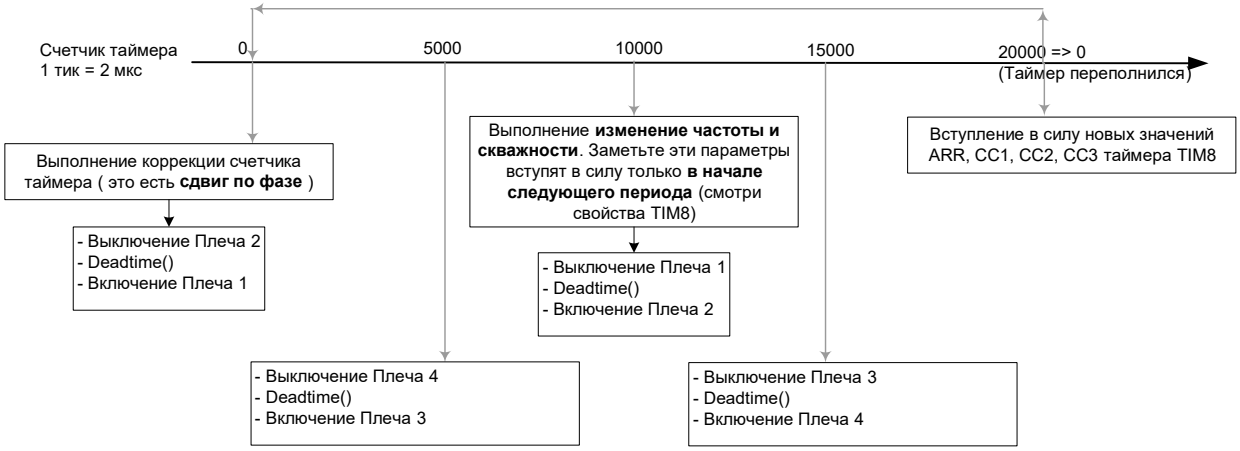


Рисунок 18 – Алгоритм управления плечами моста: частота 25 Гц, заполнение 50% (90º)

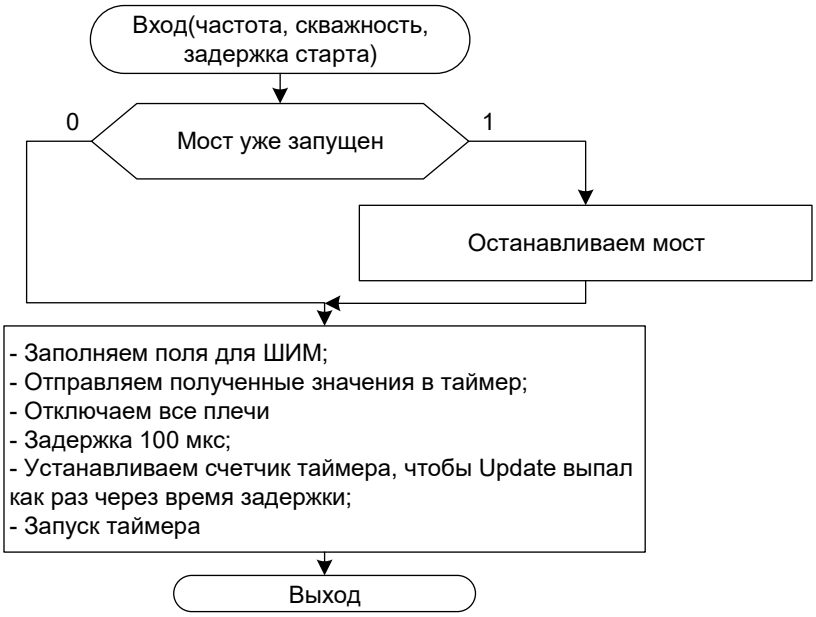


Рисунок 19 – Алгоритм запуска моста *bridge::Start()*

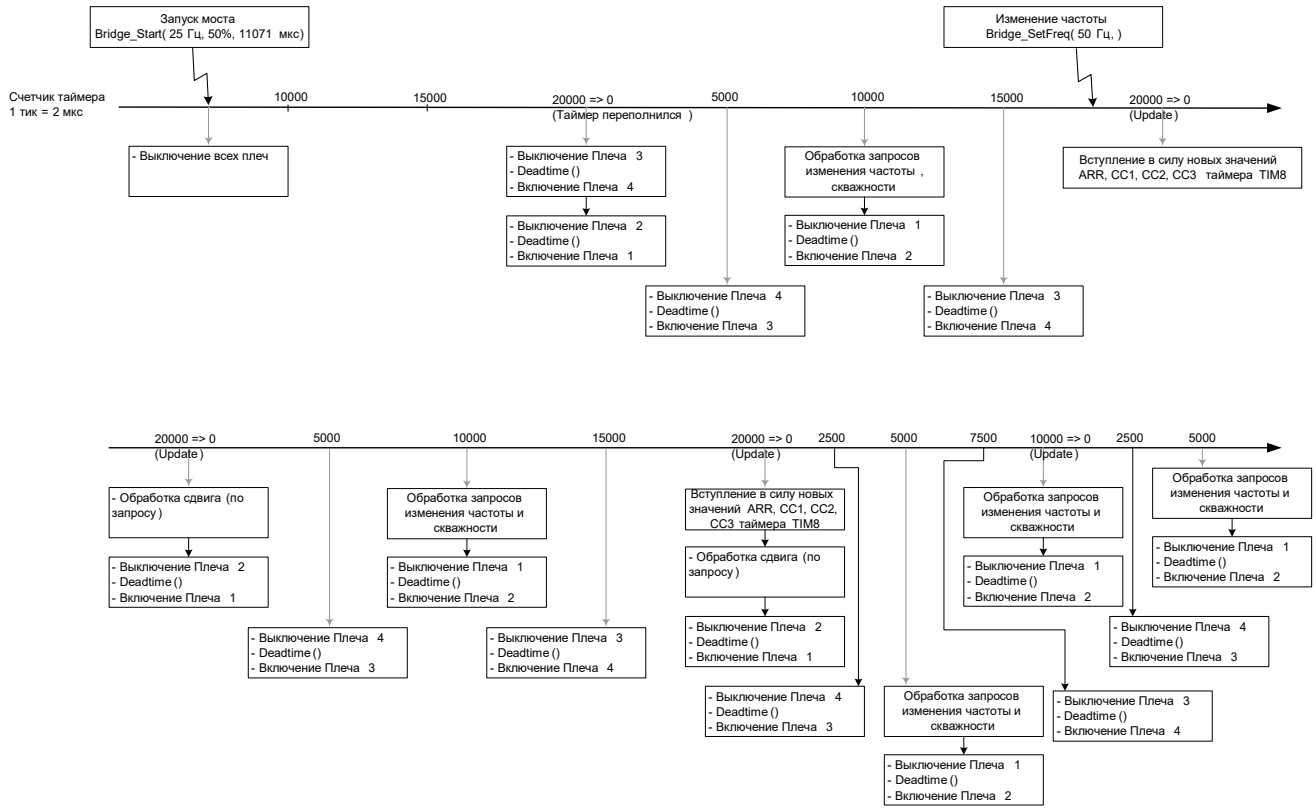


Рисунок 20 – Временной алгоритм запуска выхода моста с частотой 25 Гц, скважность 50% (90º) и задержкой в 11071 микросекунду. Затем изменение частоты на 50 Гц

3.2.2.9. Модуль измерений

Измеряет следующий параметры:

- частоту и RMS сети синхронизации,
- RMS тока выхода инвертора,
- температуру,
- питание моста 48В,
- напряжения открытия ключей.

Соответствующие параметры можно получить с помощью функции *measure::GetValue()*. Типы параметров представлены в перечислении *TypeMeasure*.

Измерения заполняются в буфер *m_bufAdc_* с использование DMA в цикличном режиме, то есть без остановки. На половине заполнения и полном заполнении происходит прерывание, которое разблокирует задачу измерений.

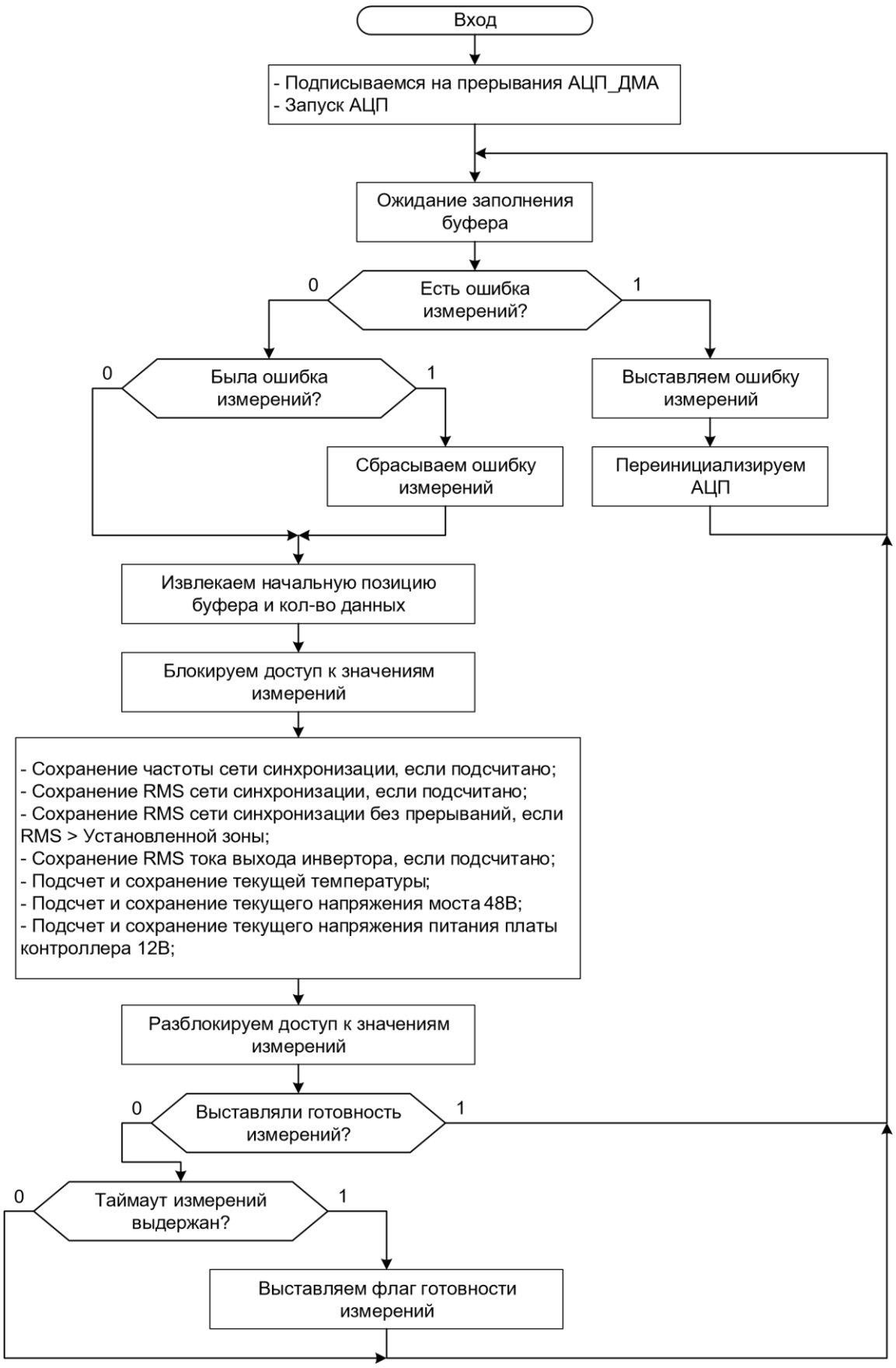


Рисунок 21 – Алгоритм задачи измерения *Measure_Task()*

4. Связь со сторонним ПО

Взаимодействие с другими программами осуществляется по следующим протоколам связи в соответствии со стандартом Modbus RTU.

«G119_GCC» обеспечивает получение через цифровой интерфейс следующей информации:

- Напряжение питания;
- Выходной ток инвертора;
- Частота на выходе устройства;
- Температура внутри устройства;
- Синхронизация с сетью;
- Результат самодиагностики;
- Последняя зафиксированная неисправность;
- Состояние связи с терминалом;
- Серийный номер;
- Версия прошивки устройства;
- Время и дата.

5. Используемые технические средства

«G119_GCC» применяется в составе генератора контрольного тока типа Г119х производство ООО НПП «ЭКРА». «G119_GCC» записывается в постоянное запоминающее устройство ГКТ при их изготовлении в ООО НПП «ЭКРА», и может перезаписываться в процессе проведения наладки оборудования или технического обслуживания обученным персоналом.

6. Вызов и загрузка

Запуск «G119_GCC» происходит в автоматическом режиме при подаче напряжения на клеммы питания Г119.

7. Входные и выходные данные

7.1. Входные данные

Входными данными ПО «G119_GCC» являются:

- входные аналоговые сигналы, преобразованные в цифровые сигналы посредством АЦП;
- входные дискретные сигналы, поступающие на клеммы ГКТ;
- входные данные, приходящие по протоколам Modbus RTU;
- файл программы блока «G119_GCC», для загрузки или обновления программы блока.

7.2. Выходные данные

Выходные данные ПО «G119_GCC» разделяются на:

- данные для управления физическими выходами;
- файлы, создаваемые программой;
- данные предоставляемые по протоколам связи.
- «G119_GCC» передает управляющие воздействия на:
- выходные реле для формирования сигналов управления внешними цепями;
- драйверы управления силовыми ключами для формирования контрольного тока.

«G119_GCC» реализует доступ к текущим измерениям и параметрам по следующим протоколам связи:

- Modbus RTU.

Лист регистрации изменений

Изм.	Номера листов (страниц)				Всего листов (страниц) в документе	Номер документа	Входящий номер сопровод-ительного документа и дата	Подпись	Дата
	Изменен-ных	Заменен-ных	новых	Аннули-рованных					