



СПРАВОЧНОЕ РУКОВОДСТВО  
Программируемый логический контроллер Canny 5.2

Москва 2020  
(Редакция от 01.11.2020)

## СОДЕРЖАНИЕ

<b>1 Введение .....</b>	<b>5</b>
<b>2 Общие сведения о ПЛК .....</b>	<b>6</b>
2.1 Основные определения и используемые сокращения.....	7
2.2 Что такое контроллер? .....	8
2.3 Программное обеспечение ПЛК.....	9
2.4 Как работает ПЛК.....	10
2.5 Программирование без программиста .....	11
2.6 Среда исполнения функциональных диаграмм.....	12
2.6.1 Представление функциональной диаграммы .....	12
2.6.2 Порядок исполнения.....	12
2.6.3 Доступ к ресурсам контроллера .....	13
2.7 Архитектура ПЛК.....	15
2.8 Структура программного обеспечения .....	17
<b>3 CANNY 5.2 .....</b>	<b>19</b>
3.1 Общие сведения .....	20
3.2 Устройство и принцип работы.....	23
3.2.1 Внешний вид и расположение элементов .....	23
3.2.2 Программная архитектура .....	24
3.2.3 Структура программного обеспечения.....	25
3.3 Режимы работы.....	27
3.3.1 Режим загрузки ПО .....	27
3.3.2 Автономный режим.....	27
3.3.3 Автономный режим пониженного энергопотребления.....	28
3.4 Среда исполнения функциональных диаграмм.....	29
3.4.1 Представление функциональной диаграммы .....	29
3.4.2 Порядок исполнения.....	29
3.4.3 Доступ к ресурсам контроллера .....	30
3.5 CANNY 5.2. Аналого-цифровой преобразователь (АЦП).....	33
3.5.1 Общее описание .....	33
3.5.2 Регистры драйвера .....	33
3.5.3 Примеры.....	34
3.6 CANNY 5.2. Драйвер CAN.....	35
3.6.1 Общее описание .....	35
3.6.2 Регистры драйвера.....	35

3.6.3	Примеры.....	39
3.7	CANNY 5.2. Энергонезависимая память (ЭНП).....	41
3.7.1	Общее описание.....	41
3.7.2	Регистры энергонезависимой памяти.....	41
3.7.3	Примеры.....	42
3.8	CANNY 5.2. Драйвер каналов ввода-вывода.....	44
3.8.1	Общее описание.....	44
3.8.2	Регистры драйвера.....	45
3.8.3	Режим дискретного выхода.....	48
3.8.4	Режим широтно-импульсного выхода.....	49
3.8.5	Режим дискретного входа.....	51
3.8.6	Режим входа-счетчика.....	52
3.8.7	Эквивалентные принципиальные электрические схемы.....	54
3.9	CANNY 5.2. Драйвер пульта ИК ДУ.....	59
3.9.1	Общее описание.....	59
3.9.2	Регистры драйвера пульта ИК ДУ.....	60
3.9.3	Примеры.....	61
3.10	CANNY 5.2. Системные ресурсы и режимы работы.....	63
3.10.1	Общее описание.....	63
3.10.2	Сброс контроллера.....	63
3.10.3	Встроенный светодиод контроллера.....	64
3.10.4	Режим пониженного энергопотребления.....	65
3.10.5	Изменение разрешения системного таймера.....	67
3.10.6	Фактическое время выполнения функциональной диаграммы.....	68
3.10.7	Идентификатор устройства.....	69
3.10.8	Контроль активности интерфейсов контроллера.....	70
3.10.9	Идентификатор вендора устройства.....	71
3.11	CANNY 5.2. Драйвер UART - RS232.....	72
3.11.1	Общее описание.....	72
3.11.2	Регистры драйвера.....	73
3.11.3	Работа контроллера в режиме UART.....	75
3.12	CANNY 5.2. Параметры пользовательской конфигурации.....	76
3.12.1	Общее описание.....	76
3.12.2	Регистры параметров пользовательской конфигурации.....	76
3.12.3	Пример использования параметров пользовательской конфигурации....	77
<b>4</b>	<b>Заключение.....</b>	<b>79</b>

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

CAN	—	Controller Area Network
LIN	—	Local Interconnect Network
ПЛК	—	программируемый логический контроллер
ПО	—	программное обеспечение
ПК	—	персональный компьютер
CFD	—	Sanny Functional Diagram, графический язык программирования использующийся в интегрированной среде разработки SannyLab
ШИМ	—	широтно-импульсная модуляция
ВЧ ШИМ	—	высокочастотная ШИМ

## 1 Введение

Уважаемые коллеги! Мы выбрали для создания, редактирования и публикации документации к нашим продуктам онлайн-платформу wiki. В поставку каждой новой версии CannyLab, мы помещаем избранные статьи из [онлайн-документации](#) и дополнительные материалы в виде PDF-документов. Преобразование статей из онлайн-документации в документы происходит автоматически, мы работаем над повышением качества результата этого преобразования. Вы можете воспользоваться интересующими вас документами оффлайн, найдя их в меню Справка, либо в каталоге установки CannyLab\doc, либо в папке doc\ru, находящейся на вашем персональном компьютере в месте установки программы CannyLab.

## **2      Общие сведения о ПЛК**

- Основные определения и используемые сокращения
- Что такое контроллер?
- Программное обеспечение ПЛК
- Как работает ПЛК
- Программирование без программиста

## 2.1 Основные определения и используемые сокращения

ПЛК	—	программируемый логический контроллер
ПО	—	программное обеспечение
ПК	—	программное обеспечение
CFD	—	Sanny Functional Diagram, графический язык программирования использующийся в интегрированной среде разработки SannyLab

## 2.2 Что такое контроллер?

**Контроллер** - управляющее устройство, применяемое в промышленности, на транспорте, в других отраслях и в быту, для автоматического управления оборудованием по заданному алгоритму. Контроллеры широко используются для управления технологическими процессами, поддержания физических параметров объекта управления на заданном уровне и схожих по содержанию задач.

Одним из самых перспективных типов контроллеров, на сегодняшний день, является электронный программируемый логический контроллер (ПЛК), алгоритмы работы которого описываются программно, хранятся во внутренней памяти контроллера и выполняются встроенным в ПЛК микропроцессором. Взаимодействие электронного контроллера с объектом управления происходит посредством входящих и исходящих электрических сигналов.

Программируемые логические контроллеры, широко применяемые во встраиваемых системах, системах контроля и управления, имеют относительно простую для понимания архитектуру. Конструктивно это, как правило, довольно компактное устройство, состоящее из одного или нескольких соединенных между собой электронных модулей, содержащих разъемы для подачи на контроллер питания и подключения внешних входных и выходных электрических линий, позволяющих контроллеру взаимодействовать с внешним миром.



### **2.3 Программное обеспечение ПЛК**

Современный ПЛК имеет развитый комплекс программных средств, состоящих из операционной системы ПЛК, предоставляемой производителем контроллера и сторонних или собственных программных средств, предназначенных для разработки, отладки и записи в контроллер пользовательских программ.

Операционная система, отвечающая за выполнение контроллером пользовательского приложения, обслуживает низкоуровневую систему ввода-вывода контроллера, интерфейсы передачи данных, управляет распределением памяти, режимами энергопотребления, таймерами, осуществляет обработку ошибок, позволяя пользователю, разрабатывающему приложение, полностью сосредоточиться на алгоритмической части решения прикладной задачи.

Средства разработки и отладки пользовательских программ, позволяют создавать и корректировать программы, реализующие алгоритмы работы контроллера, моделировать на ПК процесс выполнения программы контроллером, наблюдать за промежуточными результатами вычислений, а так же записывать программное обеспечение в контроллер.

## 2.4 Как работает ПЛК

Перед применением ПЛК, в него необходимо загрузить операционную систему и пользовательскую программу, разработанную для решения конкретной прикладной задачи. Все программное обеспечение ПЛК обычно располагается в энергонезависимой памяти и защищено от повреждения в случае сброса питания. Изменение кода прикладной программы в памяти ПЛК может быть выполнено пользователем многократно.

Типовая схема работы ПЛК может быть описана следующим образом.

К внешним каналам ввода контроллера подключаются датчики, к каналам вывода - исполнительные механизмы. На контроллер подается питание и его операционная система немедленно начинает циклически исполнять пользовательское приложение.

Цикл выполнения приложения состоит из следующих, последовательных этапов:

- операционная система считывает состояние каждого входного канала контроллера и записывает его во внутреннюю память контроллера;
- операционная система, последовательно, команда за командой, выполняет всё пользовательское приложение: каждая исполняемая команда приложения считывает из внутренней памяти необходимые ей данные, производит с ними вычисления и записывает результаты своих расчетов во внутреннюю память контроллера;
- операционная система получает из внутренней памяти те значения, которые необходимо отобразить на выходе контроллера, и переводит выходные каналы в соответствующее состояние, после чего, весь цикл выполнения повторяется с начала.

Рассмотренный подход к архитектуре среды исполнения пользовательской программы, дает возможность реализовать логически параллельное исполнение контроллером нескольких задач в рамках одной пользовательской программы, позволяя автоматизировать управление несколькими одновременно происходящими процессами используя единственный ПЛК.

## 2.5 Программирование без программиста

Одной из базовых идей, лежащих в основе использования ПЛК, является упрощение системы программирования и повышение наглядности языковых средств до уровня, доступного для понимания техническому специалисту хорошо знающему и непосредственно эксплуатирующему оборудование, но не обладающему специальными знаниями в области разработки программного обеспечения.

Такой специалист, получив простой и понятный инструмент выражения своих знаний об алгоритмах управления процессами, находящимися в его ведении, во многих случаях будет способен самостоятельно реализовать и отладить программу ПЛК, а при необходимости перенастроить параметры работы оборудования и своевременно изменить программу управления.

Зачастую качество программы ПЛК созданной таким специалистом оказывается выше, чем программы, написанной по его заданию профессиональным программистом не знакомым со всеми особенностями работы автоматизируемого процесса.

## 2.6 Среда исполнения функциональных диаграмм

- [Представление функциональной диаграммы](#)
- [Порядок исполнения](#)
- [Доступ к ресурсам контроллера](#)

### 2.6.1 Представление функциональной диаграммы

Созданная в **среде разработки SannyLab** графическая функциональная диаграмма, непосредственно перед записью в контроллер автоматически обрабатывается транслятором, который выполняет проверку диаграммы на непротиворечивость, определяет порядок выполнения функциональных блоков и преобразует диаграмму в исполняемый код — последовательность машинных команд АЛУ контроллеров CANNY.

### 2.6.2 Порядок исполнения

Исполняемый код диаграммы, при записи в контроллер уже содержащий системное программное обеспечение, включается в последовательность машинных команд системного ПО. Таким образом, общая последовательность команд контроллера с загруженным системным ПО и функциональной диаграммой, будет состоять из: процедуры инициализации, исполняемой однократно после каждого сброса контроллера и исполняемого кода функциональной диаграммы, обрамленного процедурами управления ресурсами контроллера, и помещенного в бесконечно исполняемый цикл – цикл выполнения диаграммы.

Некоторые драйверы, включенные в состав системного ПО контроллера, например драйвер CAN, требуют безотлагательной реакции контроллера на возникающие в процессе приема и передачи данных программные события. Программный код таких драйверов обрабатывается контроллером асинхронно, параллельно с основным потоком исполнения. На время обработки асинхронных вызовов драйверов, исполнение основного цикла выполнения диаграммы кратковременно приостанавливается.

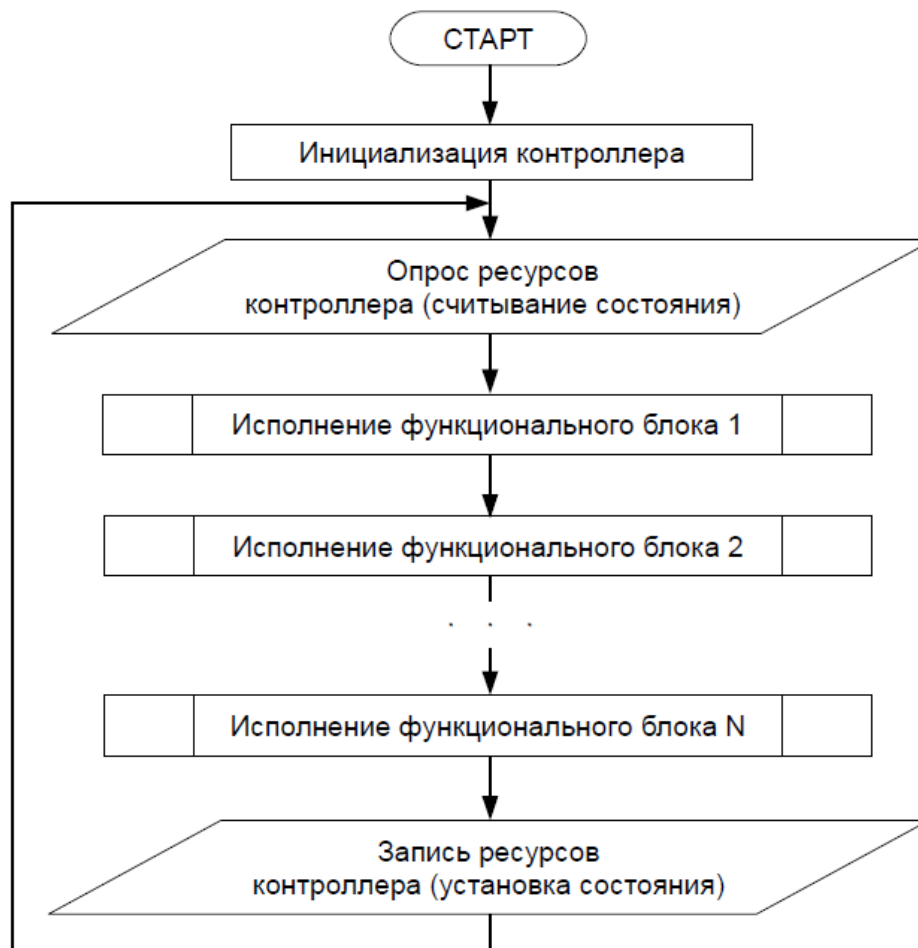


Рисунок 2.1 – Порядок исполнения

### 2.6.3 Доступ к ресурсам контроллера

Все доступные пользователю из функциональной диаграммы ресурсы: системные ресурсы контроллера, подсистема ввода-вывода и дополнительные драйверы, включенные в состав системного ПО, отображаются на защищенное адресное пространство внутренней памяти контроллера. Данное адресное пространство разделено на регистры чтения (контроля) и регистры записи:

- Регистр чтения может быть использован в качестве источника входных данных практически любого функционального блока на диаграмме и, тем самым, извлечь и использовать при реализации собственных алгоритмов сведения, полученные контроллером из внешнего мира. Например, информацию об электрическом потенциале на каком-либо контакте контроллера, или содержимое пакета данных принятого контроллером из CAN
- Регистр записи может быть использован в качестве получателя выходных данных любого функционального блока на диаграмме. Таким образом,

пользователь осуществляет управление ресурсами контроллера из функциональной диаграммы, получая возможность воздействовать на объекты внешнего мира. Например, переключить внешнее реле, изменив электрический потенциал на одном из контактов контроллера, к которому подключена его обмотка; включить контрольный светодиод; задать режим работы CAN; отправить пакет данных

Порядок использования большинства ресурсов контроллера включает в себя задание пользователем необходимых параметров их работы, например, полярности выходных каналов, полярности и чувствительности входных каналов, скорости обмена данными по CAN и т. д.

Задание таких параметров производится в форме записи специальных констант в один или в несколько определенных регистров контроллера, в зависимости от того, конфигурацию какого из ресурсов требуется задать. Например, установкой константы со значением 121 в регистр, расположенный по адресу 2432 задается режим работы канала №0 в качестве выхода положительной полярности.

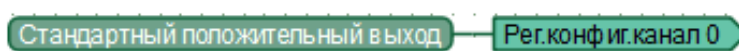


Рисунок 2.2 – Запись константы в регистр

В среде SannyLab, для удобства пользователя, все доступные регистры контроллера именованы, как и все специальные константы, использующиеся при взаимодействии с ресурсами контроллера. Поэтому для пользователя SannyLab данная операция будет выглядеть как установка константы с именем «Стандартный положительный выход» в регистр с именем «Регистр конфигурации канала №0».

Установив таким образом режим работы канала №0, мы можем по появлению значения «1» в регистре, расположенном по адресу 2465 («Регистр входного значения канала №0»), узнать о приложении положительного электрического потенциала к контакту №1 разъема X2 контроллера.

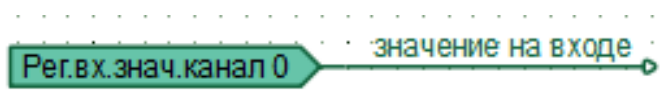


Рисунок 2.3 – Чтение данных из регистра

## 2.7 Архитектура ПЛК

Контроллеры CANNY являются цифровыми программируемыми вычислительными управляющими устройствами. В целом, для контроллеров CANNY справедливы общие сведения о программируемых логических контроллерах изложенные в статье **Общие сведения о ПЛК** данного раздела.

Основными элементами контроллеров CANNY являются:

- Арифметико-логическое устройство (АЛУ)
- Внутренняя память
- Подсистема
- Управления ходом исполнения команд
- Система ввода-вывода

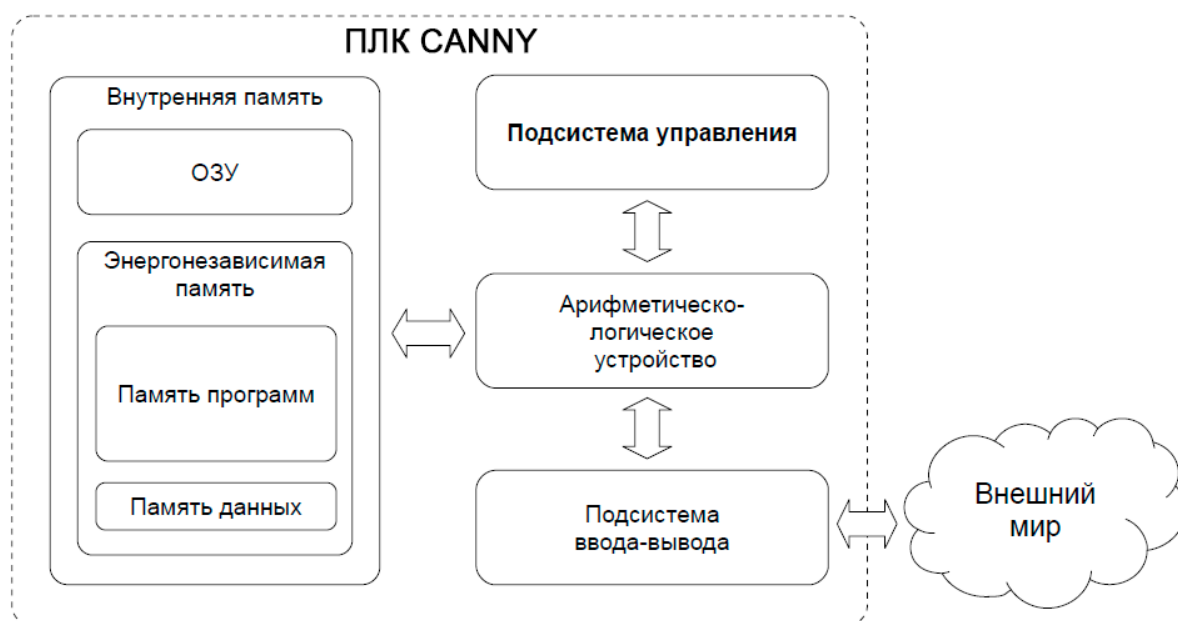


Рисунок 2.4 – Программная архитектура

Арифметико-логическое устройство — вычислительное ядро контроллеров CANNY. АЛУ обеспечивает исполнение системного программного обеспечения и пользовательских функциональных диаграмм, помещенных во внутреннюю память контроллера.

Внутренняя память контроллера разделяется на энергонезависимую память программ, энергонезависимую память данных и оперативную память данных.

Подсистема управления ходом обработки команд, отвечает за переключение и настройку режимов работы контроллера. Система ввода-вывода обеспечивает связь контроллера с внешним миром, с использованием как дискретных каналов ввода-вывода, так и стандартных цифровых интерфейсов CAN / LIN / RS232 / USB.



## 2.8 Структура программного обеспечения

Программное обеспечение CANNY состоит из:

- Программного загрузчика
- Системного ПО (операционной системы и драйверов)
- Пользовательской функциональной диаграммы

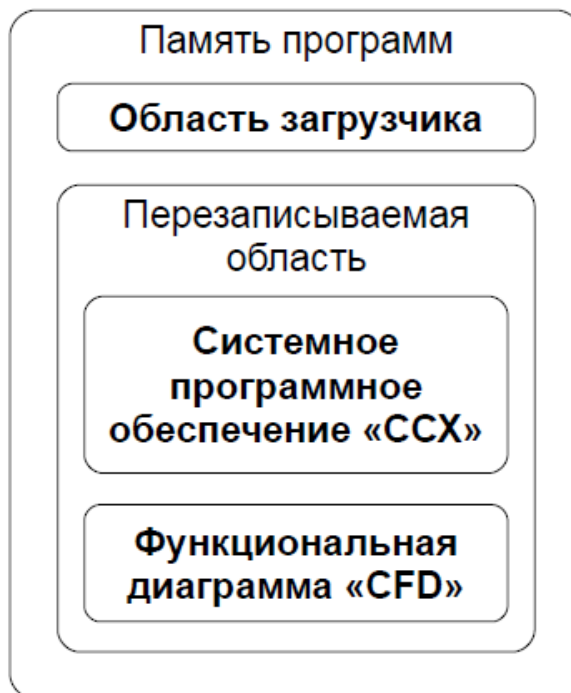


Рисунок 2.5 – Структура программного обеспечения

Программный загрузчик обеспечивает работу контроллера в режиме загрузки ПО, организуя передачу данных между контроллером CANNY и персональным компьютером по протоколу USB, осуществляет проверку целостности и запись переданного от ПК программного обеспечения во внутреннюю память контроллера. Программный загрузчик помещается во внутреннюю память контроллера в процессе его производства и не может быть удален или изменен пользователем.

Системное программное обеспечение CANNY распространяется производителем в виде файлов формата ССХ и содержит операционную систему и набор драйверов, обеспечивающих исполнение пользовательской функциональной диаграммы и её взаимодействие с ресурсами контроллера. Модификация пользователем содержимого данных файлов не допускается.

Содержимое различных файлов ССХ может быть многократно записано пользователем в контроллер.

Пользовательская функциональная диаграмма создается и модифицируется пользователем в интегрированной среде разработки SannyLab и, после записи в контроллер, задает алгоритм его работы в автономном режиме. Пользовательские диаграммы могут быть многократно записаны в контроллер и сохранены из среды SannyLab в файлы формата **CFD**.

### 3 CANNY 5.2

**CANNY 5.2** — компактный программируемый логический контроллер, ориентированный на автомобильное, бытовое и промышленное применение, является следующим поколением контроллеров CANNY 5, унаследовавших аппаратную платформу CAN-адаптеров CANNY CPLEX Plus, и представляет собой дальнейшее развитие экономичной альтернативы ПЛК CANNY 7.

- [Общие сведения](#)
- [Устройство и принцип работы](#)
- [Режимы работы](#)
- [Среда исполнения функциональных диаграмм](#)

### 3.1 Общие сведения

Также как и **CANNY 7** и **CANNY 5**, CANNY 5.2 может быть отнесен к классу интеллектуальных реле или NanoPLC. Выгодно отличаясь ценой, но обладая меньшими возможностями, по сравнению с **CANNY 7**, контроллеры CANNY 5.2, тем не менее, могут также успешно применяться для решения многих несложных задач автоматизации, контроля и управления, для которых возможности CANNY 7 были бы избыточными.

В отличие от контроллеров **CANNY 5**, CANNY 5.2 поддерживает протокол обмена данными UART и возможность работы одного из его каналов ввода-вывода в режиме аналого-цифрового преобразователя. Также контроллер имеет 4, а не 3 как у CANNY 5, канала ввода-вывода, способных работать в режиме входа.

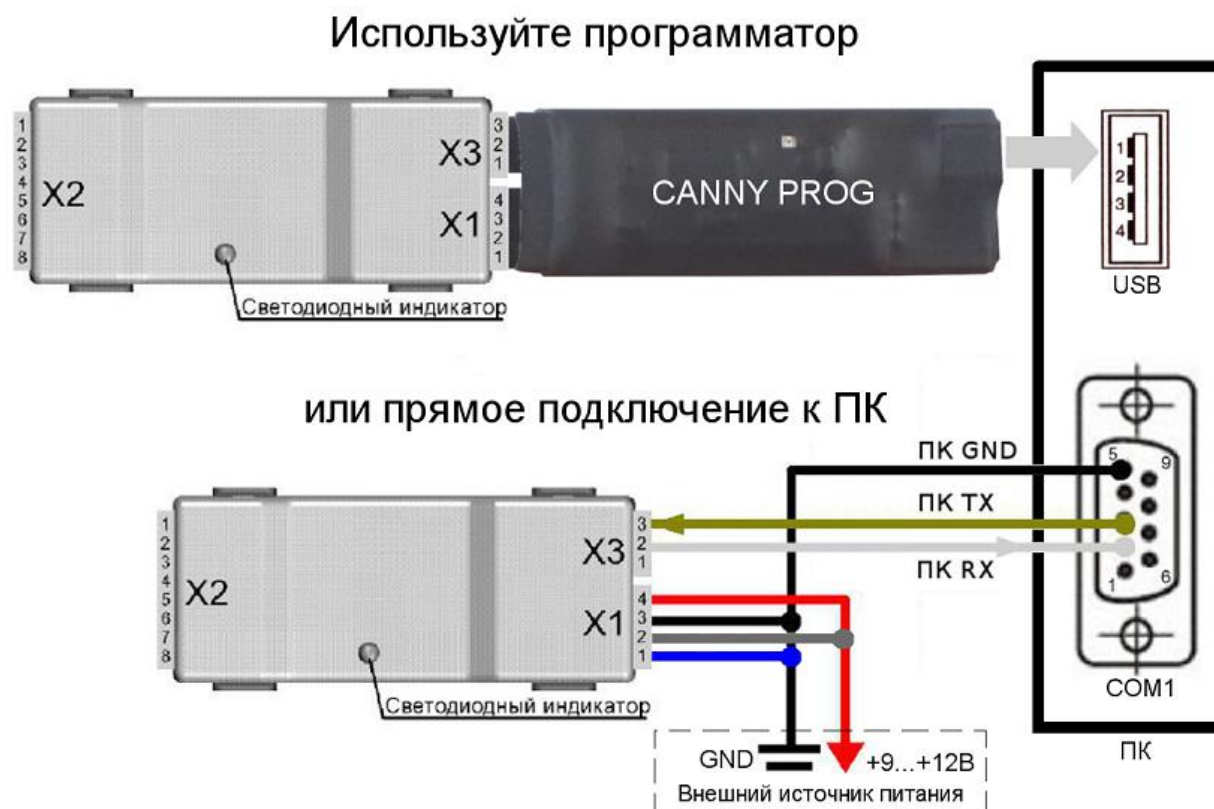
К основным особенностям CANNY 5.2 можно отнести:

- номинальное напряжение питания 9 / 18В;
- номинальное напряжение каналов ввода-вывода 0 / 12В (18В max);
- 7 выходных каналов и 4 канала ввода/вывода отрицательной полярности с максимальным током каждого из них 120мА, достаточного для управления типовыми автомобильными реле;
- интерфейс CAN 2.0В совместимый с ISO-11898, SAE J2411 широко применяемым в автомобилях;
- 11 каналов контроллера могут работать в режиме ШИМ с разрешением 100мкс, обеспечивая частоту выходного сигнала до 5кГц;
- встроенные средства управления собственным энергопотреблением контроллера в диапазоне от 5 до 30мА, позволяющие экономно расходовать заряд аккумулятора во время простоя автомобиля;
- энергонезависимая память программ и шестьдесят четыре 16-и битные ячейки энергонезависимой памяти данных доступные пользователю приложению, способные сохранить критически важные данные при сбоях питания;
- широкий диапазон рабочих температур от -40 до + 85 оС;
- встроенная защита от высоковольтных выбросов и переплюсовки питания;

— компактный корпус соответствующий классу защиты IP50 подходит для монтажа и эксплуатации в составе оборудования кабины автомобиля.

Для написания пользовательских программ CANNY 5.2 используется тот же самый графический язык программирования CFD, что применяется для программирования CANNY 7 и та же среда разработки *CannyLab*. Язык CFD позволяет быстро создавать эффективные пользовательские приложения — функциональные диаграммы, а бесплатная интегрированная среда разработки CannyLab версии 1.0 или выше, содержит средства редактирования, отладки и записи программного обеспечения в CANNY 5.2.

Для записи программного обеспечения в контроллер требуется либо специальный программатор CANNY PROG, подключаемый к ПК через порт USB, либо использование прямого подключения контроллера к COM-порту ПК.



Доступный пользователю объем памяти контроллера способен вместить программы, состоящие из нескольких сотен функциональных блоков, что позволяет реализовать достаточно сложные алгоритмы.

Контроллер имеет 11 внешних каналов, режим работы которых может быть настроен индивидуально из пользовательской функциональной диаграммы.

Каналы №№0, 1, 3, 4, 5, 6, 7 являются каналами вывода, каналы №№2, 8, 9, 10 могут функционировать как в режиме ввода так и в режиме вывода.

Светодиодный индикатор, управляемый из пользовательского приложения удобен для индикации режимов работы контроллера и диагностики.

## 3.2 Устройство и принцип работы

### 3.2.1 Внешний вид и расположение элементов

Основными конструктивными элементами CANNY 5.2 являются: микроконтроллер (MCU) со вспомогательными цепями, система электропитания всех элементов контроллера, схема согласования электрических уровней каналов ввода-вывода, система электрической защиты, разъемы и индикаторный светодиод, размещенные на единой печатной плате 65 x 23 мм установленной внутри быстроразборного пластикового корпуса.

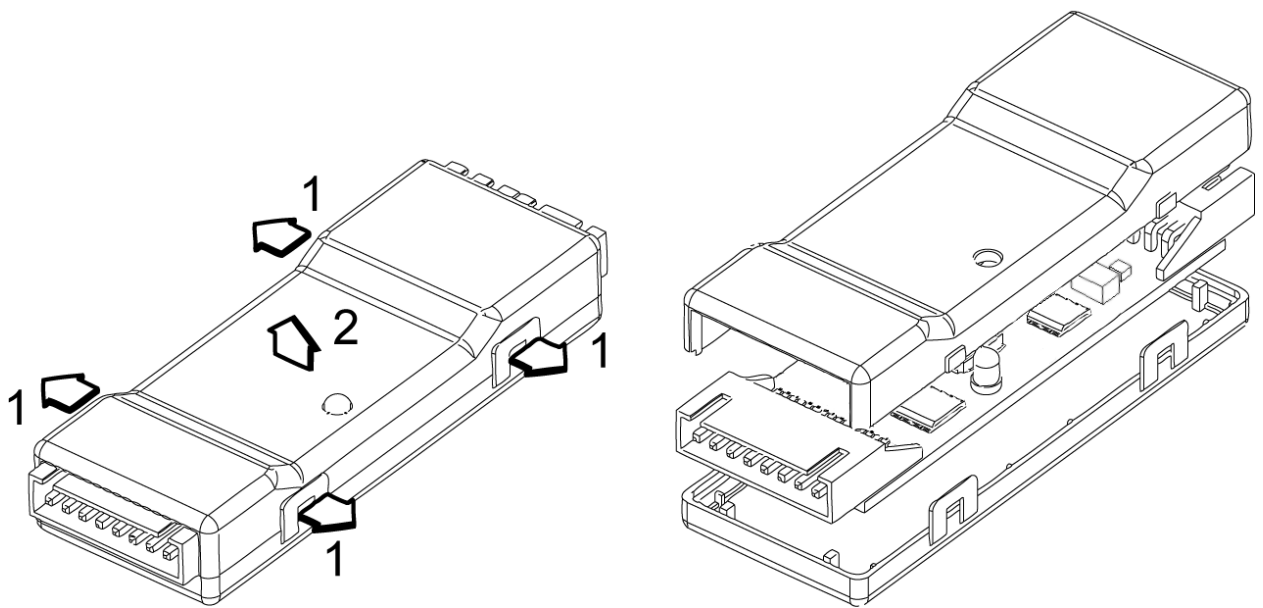


Рисунок 3.2 – Корпус CANNY 5.2

Контроллер имеет три наружных разъема. Для подключения контроллера к питанию и внешним устройствам, в комплект его поставки включен набор соединительных жгутов.

Наружный разъем X1 содержит четыре контакта: вход питания +12В, вход питания GND, CAN-H и CAN-L.

Наружный разъем X2 содержит восемь контактов, соответствующих первым восьми каналам контроллера, начиная с канала №0 и заканчивая каналом №7.

Наружный разъем X3 содержит три контакта, соответствующих каналам №8, №9 и №10 контроллера.

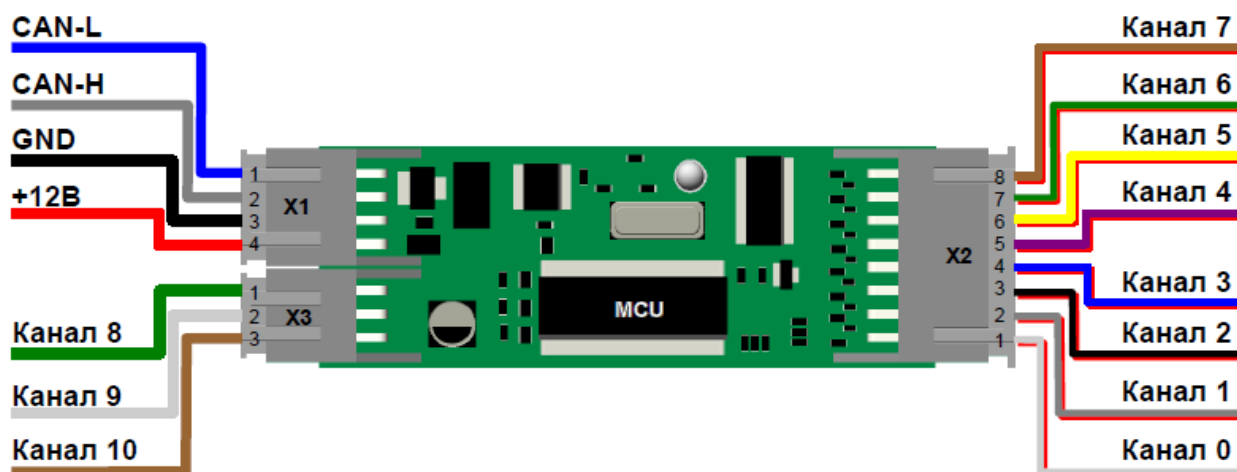


Рисунок 3.3 – Схема контроллера CANNY 5.2

### 3.2.2 Программная архитектура

CANNY 5.2 является цифровым программируемым вычислительным управляющим устройством.

В целом, для CANNY 5.2 справедливы общие сведения о программируемых логических контроллерах изложенные в статье [программируемый логический контроллер](#).

Основными элементами CANNY 5.2 являются: арифметическо-логическое устройство (АЛУ), внутренняя память, подсистема управления ходом исполнения команд и система ввода-вывода.

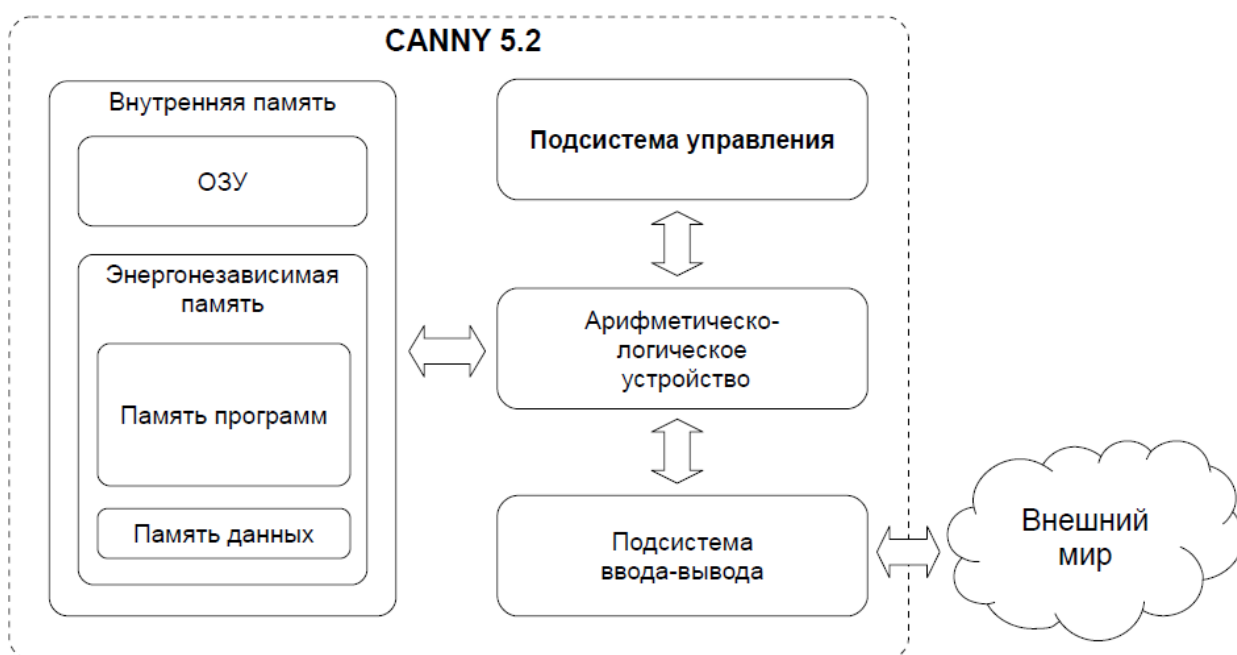


Рисунок 3.4 – Программная архитектура контроллера CANNY 5.2



Арифметическо-логическое устройство — вычислительное ядро CANNY 5.2. АЛУ обеспечивает исполнение системного программного обеспечения и пользовательских функциональных диаграмм, помещенных во внутреннюю память контроллера.

Внутренняя память контроллера разделяется на энергонезависимую память программ, энергонезависимую память данных и оперативную память данных.

Подсистема управления ходом обработки команд, отвечает за переключение и настройку режимов работы контроллера.

Система ввода-вывода обеспечивает связь контроллера с внешним миром, с использованием как дискретных каналов ввода-вывода, так и стандартного цифрового интерфейса CAN.

### 3.2.3 Структура программного обеспечения

Программное обеспечение CANNY 5.2 состоит из: программного загрузчика, системного ПО (операционной системы) и пользовательской функциональной диаграммы.



Рисунок 3.5 – Структура программного обеспечения контроллера CANNY 5.2

Программный загрузчик обеспечивает работу контроллера в режиме загрузки ПО, обеспечивая передачу данных между CANNY 5.2 и персональным компьютером по RS-232, осуществляет проверку целостности и запись переданного от ПК программного обеспечения во внутреннюю память контроллера. Программный загрузчик помещается во внутреннюю память контроллера в процессе его производства и не может быть удален или изменен пользователем.

Системное программное обеспечение CANNY 5.2 распространяется производителем в виде файлов формата CCX и содержит операционную систему и набор драйверов, обеспечивающих исполнение пользовательской функциональной диаграммы и её взаимодействие с ресурсами контроллера. Модификация пользователем содержимого данных файлов не допускается. Содержимое различных файлов CCX может быть многократно записано пользователем в контроллер.

Пользовательская функциональная диаграмма создается и модифицируется пользователем в интегрированной среде разработки CannyLab и, после записи в контроллер, задает алгоритм его работы в автономном режиме. Пользовательские диаграммы могут быть многократно записаны в контроллер и сохранены из среды CannyLab в файлы формата CFD.

### 3.3 Режимы работы

Предусмотрено несколько режимов работы контроллера, предназначенных для выполнения основных операций с ним.

#### 3.3.1 Режим загрузки ПО

В данном режиме, контроллер функционирует под управлением встроенного программного загрузчика, выполняющего запись системного программного обеспечения и функциональной диаграммы в контроллер по командам CannyLab. Вход в режим осуществляется при подключении контроллера к ПК с помощью специального программатора или в соответствии со схемой подключения к ПК из раздела [“Общие сведения о контроллере”](#)

Для перехода контроллера в данный режим необходимо подключить его к программатору, при этом включается встроенный зеленый светодиод контроллера, и установить соединение устройства с ПК, контрольный светодиод останется включенным

*Примечание: Перед подключением контроллера программатор должен быть уже подключен к ПК. Для корректной работы программатора с операционной системой ПК может потребоваться установка специального [драйвера](#), доступного для бесплатной загрузки. Особенности подключения контроллера к ПК приведены в разделе ["Работа с контроллером"](#)*

Выход из данного режима происходит автоматически, при разрыве соединения контроллера с ПК. Если в момент выхода из режима загрузки ПО, энергонезависимая память программ контроллера содержала корректно записанное системное программное обеспечение, то при очередном подключении питания контроллер переходит в автономный режим работы.

#### 3.3.2 Автономный режим

Автономный режим является основным режимом работы контроллера. В данном режиме контроллер под управлением загруженного в него системного программного обеспечения последовательно, в бесконечном цикле, исполняет функциональную диаграмму, работая по алгоритму заданному пользователем.

Переход в данный режим происходит автоматически, при подключении контроллера к внешнему питанию 12В, в отсутствие посторонних потенциалов на выводах CAN-L и CAN-H контроллера.

При работе в данном режиме, функциональной диаграмме пользователя доступны все ресурсы контроллера, драйверы которых включены в загруженное системное программное обеспечение.

### **3.3.3 Автономный режим пониженного энергопотребления**

Данный режим является вариантом обычного автономного режима, в котором после каждого цикла исполнения функциональной диаграммы, контроллер делает паузу в работе, снижая своё энергопотребление до минимального. Таким образом, контроллер работает в пульсирующем режиме, периодически «засыпая» и «просыпаясь».

Включением, отключением и настройкой параметров данного режима управляет функциональная диаграмма.

Использование данного режима актуально при разработке систем, ориентированных на батарейное питание, таких как бортовое автомобильное оборудование.

### **3.4 Среда исполнения функциональных диаграмм**

#### **3.4.1 Представление функциональной диаграммы**

Созданная в среде CannyLab графическая функциональная диаграмма, непосредственно перед записью в контроллер автоматически обрабатывается транслятором, который выполняет проверку диаграммы на непротиворечивость, определяет порядок выполнения функциональных блоков и преобразует диаграмму в исполняемый код — последовательность машинных команд АЛУ контроллера CANNY 5.2.

#### **3.4.2 Порядок исполнения**

Исполняемый код диаграммы, при записи в контроллер, память которого уже содержит системное программное обеспечение, включается в последовательность машинных команд системного ПО. Таким образом, общая последовательность команд контроллера с загруженным системным ПО и функциональной диаграммой, будет состоять из: процедуры инициализации, исполняемой однократно после каждого сброса контроллера и исполняемого кода функциональной диаграммы, обрамленного процедурами управления ресурсами контроллера, и помещенного в бесконечно исполняемый цикл – *цикл выполнения диаграммы*.

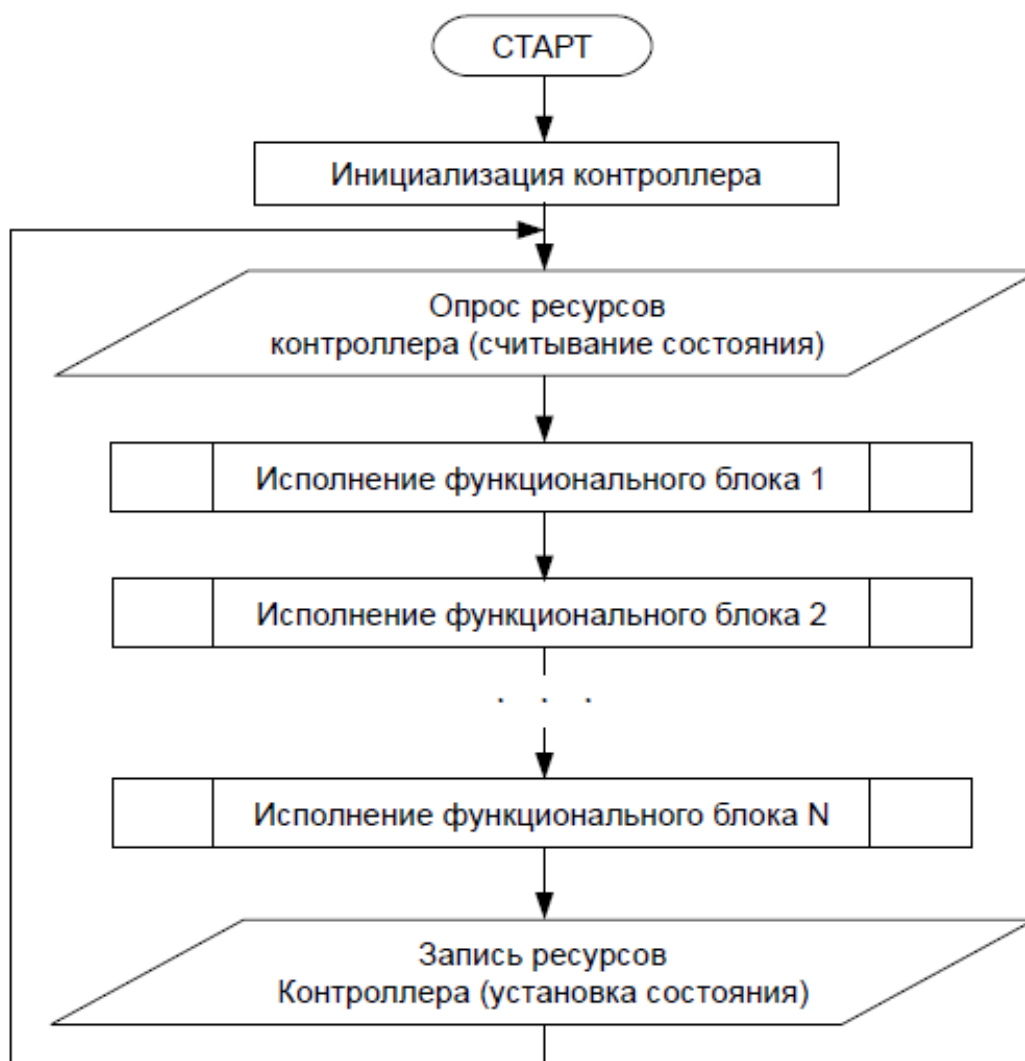


Рисунок 3.6 – Алгоритм исполнения диаграммы контроллером CANNY 5.2

Некоторые драйверы, включенные в состав системного ПО контроллера, например драйвер CAN, требуют безотлагательной реакции контроллера на возникающие в процессе приема и передачи данных программные события. Программный код таких драйверов обрабатывается контроллером асинхронно, параллельно с основным потоком исполнения. На время обработки асинхронных вызовов драйверов, исполнение основного цикла выполнения диаграммы временно приостанавливается.

### 3.4.3 Доступ к ресурсам контроллера

Все доступные пользователю из функциональной диаграммы ресурсы: системные ресурсы контроллера, подсистема ввода-вывода и дополнительные драйверы включенные в состав системного ПО, отображаются на защищенное адресное пространство внутренней памяти контроллера. Данное адресное пространство разделено на регистры чтения (контроля) и регистры записи.

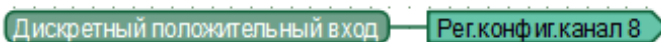
Пользователь имеет возможность указать регистр чтения в качестве источника входных данных практически любого функционального блока на диаграмме и, тем самым, извлечь и использовать при реализации собственных алгоритмов сведения, полученные контроллером из внешнего мира. Например информацию об электрическом потенциале на каком-либо контакте контроллера, или содержимое пакета данных принятого контроллером из CAN.

Регистр записи может быть использован в качестве получателя выходных данных любого функционального блока на диаграмме. Таким образом, пользователь осуществляет управление ресурсами контроллера из функциональной диаграммы, получая возможность воздействовать на объекты внешнего мира. Например, переключить внешнее реле, изменив электрический потенциал на одном из контактов контроллера, к которому подключена его обмотка; включить контрольный светодиод; задать режим работы CAN; отправить пакет данных.

Порядок использования большинства ресурсов контроллера включает в себя задание пользователем необходимых параметров их работы, например полярности выходных каналов, полярности и чувствительности входных каналов, скорости обмена данными по CAN и т.д.

Задание таких параметров производится в форме записи специальных констант в один или в несколько определенных регистров контроллера, в зависимости от того, конфигурацию какого из ресурсов требуется задать. Например, передачей константы со значением 19 в регистр, расположенный по адресу 1082 задается режим работы канала №8 в качестве входа положительной полярности.

В среде CannyLab, для удобства пользователя, все доступные регистры контроллера поименованы, как и все специальные константы, использующиеся при взаимодействии с ресурсами контроллера. Поэтому для пользователя CannyLab данная операция будет выглядеть как установка константы с именем «Дискретный положительный вход» в регистр с именем «Регистр конфигурации канала №8».



Дискретный положительный вход — Рег.конфиг.канал 8

Рисунок 3.7 – Запись константы в регистр контроллера CANNY 5.2

Установив таким образом режим работы канала №8, мы можем по появлению значения «1» в регистре расположенном по адресу 1115 («Регистр входного значения канала №8»), узнать о приложении положительного электрического потенциала к контакту №1 разъема X3 контроллера.



Рисунок 3.8 – Получение сигнала на выходе регистра контроллера CANNY 5.2



### 3.5 CANNY 5.2. Аналого-цифровой преобразователь (АЦП)

- [Общее описание](#)
- [Регистры драйвера](#)
- [Примеры](#)

#### 3.5.1 Общее описание

Один из одиннадцати каналов ввода-вывода CANNY 5.2, а именно канал №10, может работать в качестве аналого-цифрового преобразователя и использоваться для изменения напряжений от 0 до 15В.

Для активации драйвера АЦП необходимо передать значение, не равное «0», в «Регистр включения АЦП канала №10».

***Примечание:** Активация канала для работы в качестве аналого-цифрового преобразователя автоматически переводит данный канал в режим преобразования аналоговых сигналов, делая невозможным его использование в качестве дискретного входа или выхода драйвера ввода-вывода, т. к. драйвер АЦП имеет приоритет, при этом в регистрах входных и выходных значений данных каналов будут находиться результаты работы драйвера АЦП.*

АЦП канала №10 контроллера имеет разрешение 12 бит.

#### 3.5.2 Регистры драйвера

Ниже приведено описание допустимых значений регистров управления работой драйвера АЦП.

Регистры установки конфигурации драйвера АЦП позволяют включить соответствующий канал для работы в качестве аналого-цифрового преобразователя:

Таблица 3.1 – Допустимые значения установки регистра включения аналого-цифрового преобразователя канала №10.

Регистр	Ожидаемые значения
Регистр включения аналого-цифрового преобразователя канала №10	$\geq 1$ = активировать драйвер АЦП; 0 = деактивировать драйвер АЦП (канал находится под управлением драйвера ввода-вывода и доступен для использования в качестве дискретного входа/выхода).

Регистры контроля драйвера АЦП содержат информацию о состоянии канала. Ниже приведено описание возвращаемых значений регистров контроля драйвера АЦП.

Таблица 3.2 – Значения возвращаемые регистром значения АЦП канала №10.

Регистр	Возвращаемые значения
Регистр значения АЦП канала №10	0...4095 = результат работы аналого-цифрового преобразователя (измерений напряжения аналогового сигнала) соответствующего канала АЦП.

### 3.5.3 Примеры

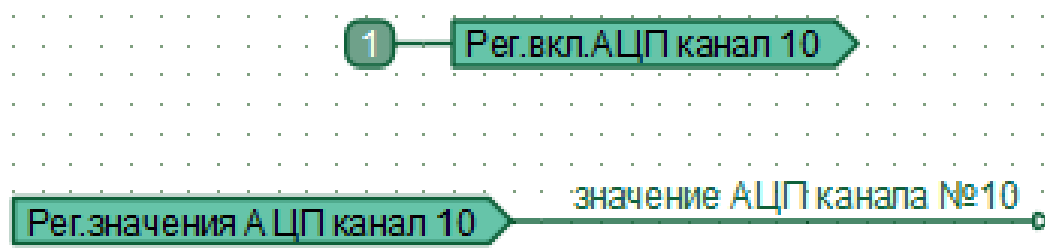


Рисунок 3.9 – Пример функциональной диаграммы активации АЦП канала №10 и получения результатов аналого-цифрового преобразования.

*Примечание:* Аналого-цифровые преобразователи контроллера CANNY 5.2 являются линейными.

*Примечание:* Измерение значений напряжения канала №10 производится контроллером один раз за 1 мс. В промежутке между измерениями регистр значения АЦП соответствующего канала сохраняет предыдущее измеренное значение.

### 3.6 CANNY 5.2. Драйвер CAN

- [Общее описание](#)
- [Регистры драйвера](#)
- [Примеры](#)

#### 3.6.1 Общее описание

Два специальных контакта контроллера CANNY 5.2, CAN-H и CAN-L, предназначены для подключения к цифровой информационной **шине «CAN»**.

Особенностью драйвера CAN контроллера CANNY 5.2 является то, что у него количество фильтров принимаемых сообщений CAN равно 8, против 16 у CANNY 7.

#### 3.6.2 Регистры драйвера

Ниже приведено описание допустимых значений регистров управления работой драйвера CAN.

Регистры конфигурации драйвера CAN позволяют установить параметры работы контроллера в качестве узла шины CAN:

Таблица 3.3 – Допустимые значения регистров управления работой драйвера CAN.

Регистр	Ожидаемые значения
Регистр конфигурации CAN	<b>1...N</b> = активизация драйвера и установка скорости приема / передачи CAN-сообщений (задается специальной константой из справочника констант); <b>0</b> = драйвер отключен.
Регистр установки фильтра CAN IDL №0 ... CAN IDL №7	<b>0...0xFFFF</b> = установить значение фильтра для младшей части идентификатора CAN-сообщения (биты 0...10 идентификатора стандартного формата или биты 0...15 идентификатора расширенного формата);
Регистр установки фильтра CAN IDH №0 ... CAN IDH №7	<b>0...0x1FFF</b> = установить значение фильтра для старшей части идентификатора CAN-сообщения (биты 16...28 идентификатора расширенного формата);
Регистр установки режима пассивного приема сообщений CAN	<b>≥ 1</b> = включен режим пассивного приема (listen only) сообщений CAN; <b>0</b> = включен режим нормального приема-передачи (normal) сообщений CAN

Регистр режима фильтрации приема данных CAN	$\geq 1$ = режим фильтрации принимаемых сообщений CAN включен; <b>0</b> = режим фильтрации принимаемых сообщений CAN отключен.
Регистр запрета автоматической повторной отправки сообщения CAN при ошибке	$\geq 1$ = режим запрета автоматической повторной отправки сообщения CAN при ошибке включен, драйвером будет предприниматься только одна попытка отправить данные CAN; <b>0</b> = режим запрета автоматической повторной отправки сообщения CAN при ошибке отключен, отправка сообщений будет выполняться согласно стандарту CAN - до получения подтверждения о успешном приеме хотя бы от одного узла сети.

**Примечание:** В режиме пассивного приема сообщений CAN (*listen only*) в отличие от нормального режима CAN (*normal*) драйвер выполняет прием данных из CAN-шины, но при этом не отправляет подтверждение их приема и не переводит сеть в состояние ошибки при обнаружении таковой. Таким образом контроллер остается незаметным для остальных устройств на шине, никак себя не проявляя. Для нормальной работы сети, в ней должны находиться минимум два устройства работающие в режиме *normal*. При активированном пассивном режиме приема сообщений CAN, отправка пользователем сообщений в CAN-шину также невозможна.

**Примечание:** При включенном режиме фильтрации CAN драйвер будет принимать только те сообщения, идентификаторы которых совпадают с указанными в регистрах установки фильтра приема сообщений CAN значениями, игнорируя все остальные.

**Примечание:** В режиме запрета автоматической повторной отправки сообщения CAN при ошибке драйвер выполняет однократную попытку отправки данных в CAN-шину, не "засоряя" сеть неполученными сообщениями, избегая ее зависания. При этом доставка сообщения хотя бы одному получателю не гарантируется.

Конфигурация драйвера CAN, определяется константой, задающей скорость приема/передачи данных.

Таблица 3.4 – Значения константы скорости приема/передачи данных.

Параметр	Перечень допустимых значений
Скорость приема/передачи данных, кбит/с	10; 20; 33.3; 50; 83; 95.2; 100; 125; 250; 500; 1000

Именованные константы, определяющие конфигурацию CAN-драйвера, содержатся в разделе «Конфигурация CAN» справочника констант CannyLab, доступ к которому осуществляется через контекстное меню входа функционального блока, имеющего тип «Константа».

Регистры диагностики драйвера CAN позволяют пользователю определить состояние драйвера в тот или иной момент выполнения диаграммы.

Таблица 3.5 – Регистры диагностики драйвера CAN.

Регистр	Возвращаемые значения
Регистр отсутствия активности драйвера CAN	<b>1</b> = активность драйвера CAN отсутствует, шина бездействует, прием данных не осуществляется; <b>0</b> = регистрируется активность CAN.
Регистр переполнения буфера приема CAN	<b>1</b> = ошибка, буфер CAN переполнен; <b>0</b> = переполнение буфера приема отсутствует.
Регистр ошибки приема / передачи CAN	<b>1</b> = уровень ошибок приема / передачи CAN превысил допустимый порог; <b>0</b> = уровень ошибок приема / передачи ниже допустимого порога.
Регистр готовности буфера передачи данных CAN	<b>1</b> = буфер передачи данных драйвера CAN свободен и готов к загрузке новых сообщений; <b>0</b> = буфер передачи данных драйвера не готов.

Регистры приема драйвера CAN позволяют получить доступ к значениям, полученным по шине.

Таблица 3.6 – Регистры приёма драйвера CAN.

Регистр	Возвращаемые значения
Регистр наличия принятых данных CAN	<b>1</b> = в буфере приема драйвера CAN находится полученное сообщение, данное значение появляется в регистре на один цикл выполнения диаграммы сообщая об актуальности данных, находящихся в буфере приема; <b>0</b> = в буфере приема драйвера CAN нет актуальных данных.
Регистр принятого сообщения CAN IDL	<b>0...0xFFFF</b> = значение младшей части идентификатора полученного CAN-сообщения
Регистр принятого сообщения CAN IDH	<b>0...0x1FFF</b> = значение старшей части идентификатора полученного CAN-сообщения.

Регистр принятого сообщения CAN ERL	<b>0...0xC000</b> = значение, равное количеству байт данных в принятом сообщении, признаки EXT и RTR (см. примечание).
Регистр принятого сообщения CAN D1:D0 ... CAN D7:D6	<b>0...0xFFFF</b> = значения соответствующих байт данных принятого сообщения CAN, по два байта на регистр.

***Примечание:** Регистр принятого сообщения CAN ERL, помимо числа байт в принятом сообщении 0...8 в младших битах, содержит в своих старших битах информацию о специальных признаках сообщения: бит 15 - признак EXT и бит 14 признак RTR. Где EXT = 1 при приеме сообщения в расширенном формате, EXT = 0 при стандартном формате сообщения; RTR = 1 при приеме удаленного запроса данных, EXT = 0 при приеме обычного сообщения.*

Регистры передачи сообщений CAN используются для размещения в буфере передачи драйвера данных, подлежащих отправке.

Таблица 3.7 – Регистры передачи сообщений CAN.

Регистр	Ожидаемые значения
Регистр начала передачи CAN	$\geq 1$ = загрузить данные из регистров передачи в буфер передачи драйвера CAN; 0 = не загружать данные в буфер передачи драйвера CAN.
Регистр сообщения передачи CAN IDL	<b>0...0xFFFF</b> = значение младшей части идентификатора передаваемого CAN-сообщения.
Регистр сообщения передачи CAN IDH	<b>0...0x1FFF</b> = значение старшей части идентификатора передаваемого CAN-сообщения.
Регистр сообщения передачи CAN ERL	<b>0...0xC000</b> = значение, равное количеству байт данных в принятом сообщении, признаки XT и RTR (см. примечание).
Регистр сообщения передачи CAN D1:D0 ... CAN D7:D6	<b>0...0xFFFF</b> = значения соответствующих байт данных принятого сообщения CAN, по два байта на регистр.

**Примечание:** Регистр сообщения передачи CAN ERL, помимо числа байт в передаваемом сообщении 0...8 в младших битах, содержит в своих старших битах информацию о специальных признаках сообщения: бит 15 - признак EXT и бит 14 признак RTR. Где EXT = 1 при передаче сообщения в расширенном формате, EXT = 0 при стандартном формате сообщения; RTR = 1 при передаче удаленного запроса данных, EXT = 0 при передаче обычного сообщения.

### 3.6.3 Примеры

Пример функциональной диаграммы отправки данных в шину CAN. Выполняя диаграмму контроллер, с периодичностью 1 раз в 100мс, передает в шину на скорости 125 кБод данные о состоянии трех своих входов, используя сообщения стандартного формата с идентификатором 0x123, содержащие один байт данных.

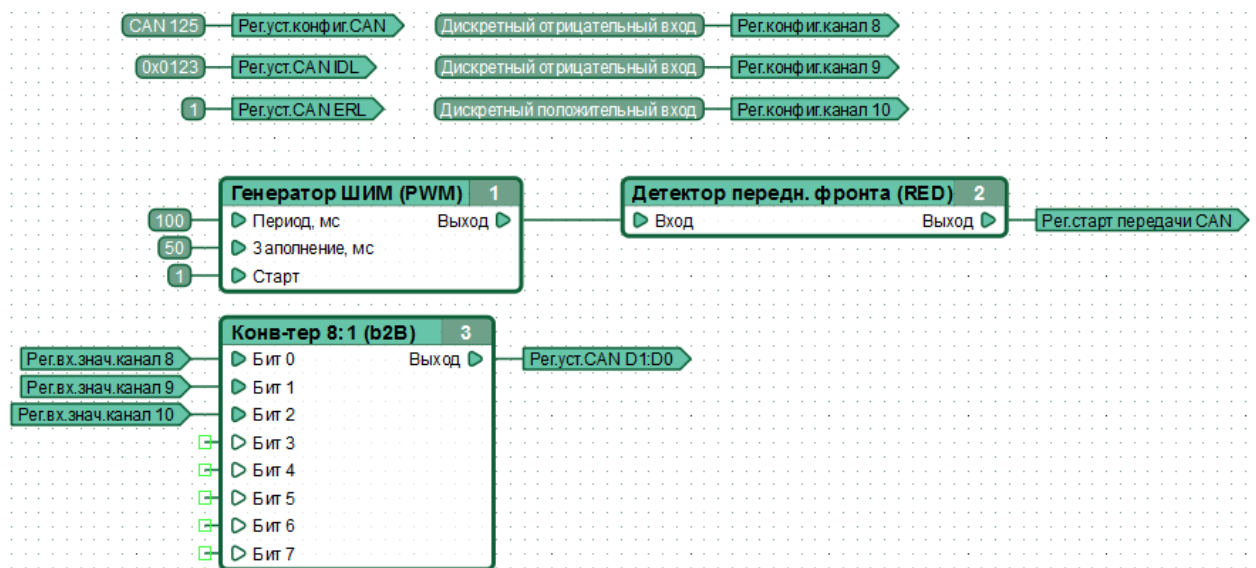


Рисунок 3.10 – Пример функциональной диаграммы отправки данных в шину CAN

Выполняя диаграмму контроллер, с периодичностью 1 раз в 100мс, передает в шину на скорости 125 кБод данные о состоянии трех своих входов, используя сообщения стандартного формата с идентификатором 0x123, содержащие один байт данных.

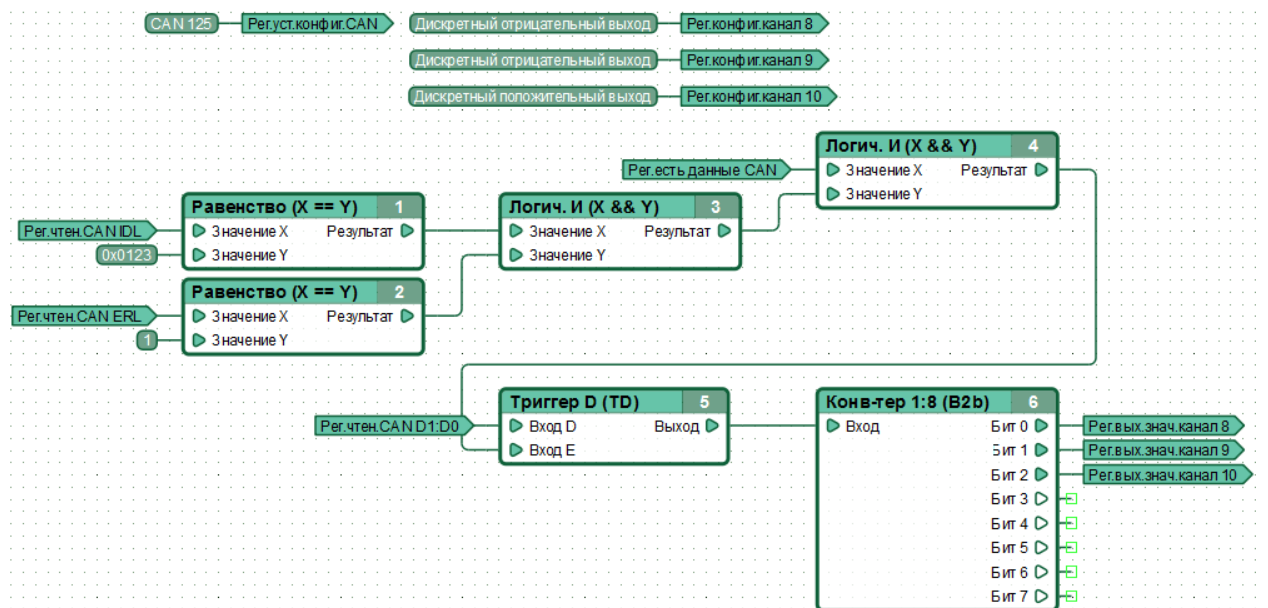


Рисунок 3.11 – Пример функциональной диаграммы приема данных из шины CAN

Выполняя диаграмму контроллер, получая по шине сообщения стандартного формата с идентификатором 0x0123, содержащие один байт данных, устанавливает на трех своих выходах состояние в соответствии с полученным значением.



### 3.7 CANNY 5.2. Энергонезависимая память (ЭНП)

- [Общее описание](#)
- [Регистры энергонезависимой памяти](#)
- [Примеры](#)

#### 3.7.1 Общее описание

Для исключения потери критически важной информации (состояния контроллера, состояния внешних устройств и т. п.) при сбросе питания, в контроллере CANNY 5.2 предусмотрено наличие энергонезависимой памяти. Сохраненные в ней значения будут доступны после восстановления питания контроллера в специальных регистрах.

Пользователю доступны 64 шестнадцатитбитные ячейки энергонезависимой памяти, доступ к которым осуществляется с помощью соответствующих регистров чтения и записи.

*Примечание: Работа с энергонезависимой памятью не требует какой-либо специальной предварительной конфигурации.*

#### 3.7.2 Регистры энергонезависимой памяти

Ниже приведено описание допустимых значений регистров установки энергонезависимой памяти контроллера. Они используются для сохранения информации в ячейках ЭНП.

Таблица 3.8 – Допустимые значения регистров установки энергонезависимой памяти контроллера.

Регистр	Ожидаемые значения
Регистр установки энергонезависимой памяти №00 ... Регистр установки энергонезависимой памяти №63	<b>0...65535</b> = сохраняемое значение

Ниже приведено описание допустимых значений регистров чтения энергонезависимой памяти контроллера. Они используются для сохранения информации в ячейках ЭНП.

Таблица 3.9 – Допустимые значения регистров чтения энергонезависимой памяти контроллера.

Регистр	Возвращаемые значения
Регистр чтения энергонезависимой памяти №00 ... Регистр чтения энергонезависимой памяти №63	0...65535 = сохраняемое значение

### 3.7.3 Примеры

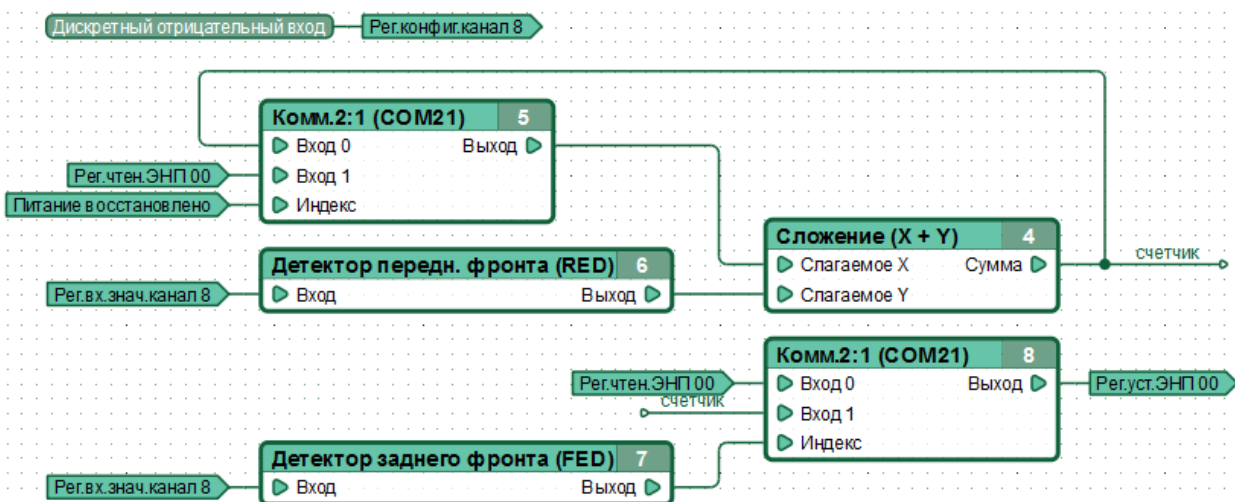


Рисунок 3.12 – Пример функциональной диаграммы работы с ячейками энергонезависимой памяти.

Количество нажатий кнопки, подключенной к каналу №8 контроллера, суммируется с ранее сохраненными в именованной сети «счетчик» значениями. Значение сети «счетчик» сохраняется в энергонезависимой памяти при отпускании кнопки. После выключения и восстановления питания контроллера, сохраненное в ячейке энергонезависимой памяти значение автоматически читается и передается обратно в именованную сеть «счетчик». Таким образом удастся избежать потери информации о количестве нажатий на данную кнопку при отключении питания контроллера.

*Примечание: Процесс сохранения данных в ЭНП требует времени, т. е. не происходит мгновенно.*

**Примечание:** Количество циклов перезаписи информации в энергонезависимой памяти ограничено. Драйвер работы с памятью CANNY 5.2 организован таким образом, что ее ресурс существенно увеличен. Тем не менее, избегайте постоянного сохранения в ЭНП ненужных данных или сохранения данных на каждом цикле выполнения диаграммы, выполняйте сохранение информации по определенному условию (смотри пример выше).

### 3.8 CANNY 5.2. Драйвер каналов ввода-вывода

*Дальнейшим развитием линейки контроллеров CANNY 5 является серия контроллеров **CANNY 5.2**. Для Ваших новых проектов рекомендуется применять актуальную версию линейки ПЛК.*

- [Общее описание](#)
- [Регистры драйвера](#)
- [Состояние канала в момент запуска контроллера](#)
- [Режим дискретного выхода](#)
- [Режим широтно-импульсного выхода](#)
- [Режим дискретного входа](#)
- [Режим входа-счетчика](#)
- [Эквивалентные принципиальные электрические схемы](#)

#### 3.8.1 Общее описание

Пользователям CANNY 5.2 доступны одиннадцать дискретных каналов контроллера: 7 выходных каналов и 4 канала ввода-вывода общего назначения. Каждый канал физически представлен соответствующим контактом разъема X2 (Каналы №№0..7) либо разъема X3 (Каналы №№8, 9 и 10) контроллера. Записывая и считывая данные соответствующих регистров драйвера, функциональная диаграмма может как управлять электрическим потенциалом на каждом из этих контактов так и получать информацию о текущем значении потенциала каждого из них.

Физические характеристики каналов позволяют подключать к ним различные внешние исполнительные устройства — электромагнитные реле, небольшие электродвигатели, светодиоды, слаботочные цепи управления оборудованием. В качестве внешних источников дискретных сигналов способных управлять работой контроллера, возможно использовать механические, электромеханические и электронные кнопки и переключатели, генераторы импульсов, источники напряжения 0-12В, транзисторные выходы различной аппаратуры и т.п.

Режим и параметры работы любого из каналов задаются функциональной диаграммой. В каждый момент времени канал может работать только в одном из возможных режимов, однако допускается динамическое переопределение конфигурации канала из функциональной диаграммы в процессе ее выполнения.

**ВНИМАНИЕ!** В отличие от CANNY 7, каналы CANNY 5.2 не имеют интеллектуальной защиты от короткого замыкания или перегрузки. Цепи каналов защищены лишь токоограничительными сгораемыми резисторами (см. [схемы](#)). Во избежание выхода каналов из строя, избегайте прямого подключения каналов с включенным выходным потенциалом GND к силовым цепям положительной полярности!

### 3.8.2 Регистры драйвера

Параметры определяющие режим работы и текущее состояние каналов контроллера задаются для каждого канала независимо друг от друга. Ниже приведено описание допустимых значений регистров управления работой каналов ввода-вывода во всех основных режимах.

Таблица 3.10 – Допустимые значения регистров управления работой каналов ввода-вывода в основных режимах.

Регистр	Ожидаемые значения
Регистр установки конфигурации канала №0	<b>1...65535</b> = установить конфигурацию канала контроллера, определяющую текущий режим и параметры его работы (задается специальной константой из справочника констант); <b>0</b> = отключить силовой выходной потенциал контакта соответствующего каналу контроллера и исключить возможность изменения его состояния из функциональной диаграммы.
Регистр установки конфигурации канала №1	
Регистр установки конфигурации канала №2	
...	
Регистр установки конфигурации канала №10	

<p>Регистр установки периода канала №0  Регистр установки периода канала №1  Регистр установки периода канала №2  ...  Регистр установки конфигурации канала №10</p>	<p><b><i>В режиме входа-счетчика в (только для каналов №№ 2,8,9,10):</i></b></p> <p><b>1...65535</b> = период подсчета числа импульсов на соответствующем контакте контроллера и обновления полученного значения регистре контроля выходного значения канала;  при использовании режима увеличения разрешения системного таймера измеряется в 0,1 миллисекунды, иначе — в целых в миллисекундах;  <b>0</b> = включить канал в режиме счетчика с накоплением.</p> <p><b><i>В режиме широтно-импульсного выхода:</i></b></p> <p><b>1...65535</b> = период в миллисекундах (или сотнях микросекунд) генерируемых импульсов на соответствующем контакте контроллера;  <b>0</b> = прекратить генерацию импульсов и установить потенциал на соответствующем контакте контроллера равным потенциалу состояния «ВЫКЛ» текущей конфигурации канала.</p> <p><b><i>В режиме дискретного входа:</i></b></p> <p><b>0...65535</b> = значение игнорируется;</p> <p><b><i>В режиме дискретного выхода:</i></b></p> <p><b>0...65535</b> = значение игнорируется;</p>
--	---

<p>Регистр установки выходного значения канала №0 Регистр установки выходного значения канала №1 Регистр установки выходного значения канала №2 ... Регистр установки выходного значения канала №10</p>	<p><b>В режиме входа-счетчика в (только для каналов №№ 2,8,9,10):</b></p> <p><b>0...65535</b> = значение игнорируется.</p> <p><b>В режиме широтно-импульсного выхода:</b></p> <p><b>0...65535</b> = заполнение (скважность) в миллисекундах генерируемых импульсов на соответствующем контакте контроллера. иначе — в целых в миллисекундах;</p> <p><b>В режиме дискретного входа:</b></p> <p><b>0...65535</b> = значение игнорируется;</p> <p><b>В режиме дискретного выхода:</b></p> <p><b>≥ 1</b> = установить на соответствующем контакте контроллера электрический потенциал заданный конфигурацией данного канала для состояния «ВКЛ»; <b>0</b> = установить на соответствующем контакте контроллера электрический потенциал заданный конфигурацией данного канала для состояния «ВЫКЛ».</p>
---	--

Регистры контроля драйвера каналов ввода-вывода разделяются на содержащие информацию о состоянии драйвера ввода-вывода в целом и на содержащие информацию о состоянии каждого канала индивидуально. Ниже приведено описание возвращаемых значений регистров контроля драйвера каналов ввода-вывода во всех основных режимах работы.

Таблица 3.11 – Возвращаемые значения регистров контроля драйвера каналов ввода-вывода в основных режимах работы.

Регистр	Возвращаемые значения
<p>Регистр контроля активности ввода-вывода</p>	<p><b>1</b> = в ходе прошедшего цикла выполнения функциональной диаграммы зарегистрировано изменение электрического потенциала на каком-либо контакте контроллера соответствующем каналу, сконфигурированному как дискретный или счетчик импульсов; либо диаграммой было изменено значение регистров выходных каналов или входов-счетчиков;</p> <p><b>0</b> = за прошедший цикл выполнения функциональной диаграммы изменений на каналах контроллера или изменений значений регистров драйвера не обнаружено.</p>

Регистр контроля входного значения канала №2	<p><b>В режиме дискретного входа или выхода</b></p> <p><b>1</b> = на соответствующем контакте контроллера установился электрический потенциал соответствующий полярности данного канала в режиме «ВКЛ»;</p> <p><b>0</b> = на соответствующем контакте контроллера установился электрический потенциал не соответствующий полярности данного канала в режиме «ВКЛ».</p>
Регистр контроля входного значения канала №8	
Регистр контроля входного значения канала №9	<p><b>В режиме входа-счетчика:</b></p> <p><b>0...65535</b> = число импульсов зарегистрированных на соответствующем контакте контроллера за прошедший период времени заданной продолжительности.</p>
Регистр контроля входного значения канала №10	<p><b>В других режимах:</b></p> <p><b>0</b> = значение всегда равно 0.</p>

**3.8.2.1 Состояние канала в момент запуска контроллера**

При включении контроллера, все силовые выходные элементы каналов находятся в выключенном состоянии. На контактах контроллера соответствующих каналам 0..9 устанавливаются потенциал 12В/1мА на контакте соответствующем каналу 10 — очень слабый отрицательный потенциал («воздух»).

**3.8.3 Режим дискретного выхода**

Канал, сконфигурированный для работы в режиме дискретного выхода, устанавливает на соответствующем контакте контроллера электрический потенциал соответствующий состоянию «ВКЛ» при записи ненулевого значения в регистр выходного значения канала, и устанавливает на соответствующем контакте контроллера электрический потенциал соответствующий состоянию «ВЫКЛ» при записи значения «0» в регистр выходного значения канала.

Конфигурация канала для работы в данном режиме, определяется константой, задающей комбинацию параметров, определяющих электрический потенциал и силу тока на контакте, в положениях «ВКЛ» и «ВЫКЛ» соответствующего канала контроллера.

Таблица 3.12 – Параметры конфигурации канала для работы в режиме дискретного выхода

Параметр	Перечень допустимых значений
Тип канала	Дискретный.



Состояние «ВКЛ»	плюс (1мА); МИНУС (120мА).
Состояние «ВЫКЛ»	МИНУС (120мА); плюс (1мА).

Именованные константы, представляющие доступные пользователю комбинации параметров конфигурации каналов, содержатся в разделе «Конфигурация каналов ввода-вывода» справочника констант CannyLab, доступ к которому осуществляется через контекстное меню входа функционального блока, имеющего тип «Константа».

Для перевода канала контроллера в режим дискретного выхода, необходимо в соответствующий каналу «Регистр конфигурации канала №XX» передать значение константы, соответствующей выбранному режиму работы.

*Примечание: Для каналов 2, 8, 9, 10, работающих в конфигурации дискретного выхода, так же возможно получить значение, соответствующее фактическому текущему электрическому потенциалу на контакте данного канала, что позволяет использовать канал в режиме обратной связи.*

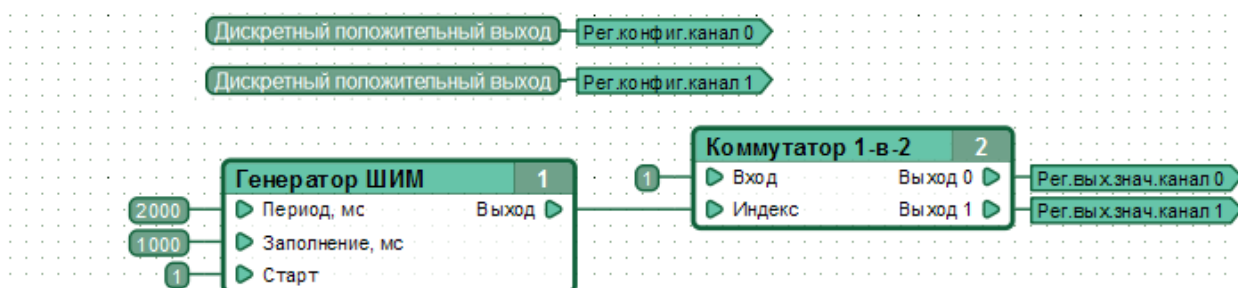


Рисунок 3.13 – Пример функциональной диаграммы выполняющей ежесекундное поочередное переключение электрических потенциалов с +12В(1мА) на GND(120мА) на контактах контроллера, соответствующих каналам №0 и №1.

**ВНИМАНИЕ!** В отличие от CANNY 7, каналы CANNY 5.2 не имеют интеллектуальной защиты от короткого замыкания или перегрузки. Цепи каналов защищены лишь токоограничительными сгораемыми резисторами (см. [схемы](#)). Во избежание выхода каналов из строя, избегайте прямого подключения каналов с включенным выходным потенциалом GND к силовым цепям положительной полярности.

#### 3.8.4 Режим широтно-импульсного выхода

Канал сконфигурированный для работы в режиме широтно-импульсного выхода, генерирует на соответствующем контакте контроллера широтно-

импульсный сигнал заданной полярности, частоты и заполнения импульсов. В данном режиме канал работает асинхронно функциональной диаграмме, что позволяет добиться большей стабильности временных параметров генерируемого сигнала, чем при организации широтно-импульсного генератора средствами функциональной диаграммы, особенно при малых значениях периода.



Рисунок 3.14 – График ШИМ

Конфигурация канала для работы в данном режиме, определяется константой, представляющей комбинацию параметров, определяющих электрический потенциал и силу тока на контакте, в активной «ВКЛ» и пассивной «ВЫКЛ» фазе генерации соответствующего канала контроллера, и парой числовых значений, определяющих частоту и заполнение генерируемых импульсов.

Таблица 3.13 – Параметры конфигурации канала для работы в режиме широтно-импульсного выхода

Параметр	Перечень допустимых значений
Тип канала	Дискретный.
Состояние «ВКЛ»	плюс (1мА); МИНУС (120мА).
Состояние «ВЫКЛ»	МИНУС (120мА); плюс (1мА).

Именованные константы, представляющие доступные пользователю комбинации параметров конфигурации каналов, содержатся в разделе «Конфигурация каналов ввода-вывода» справочника констант CannyLab, доступ к которому осуществляется через контекстное меню элемента диаграммы «Константа».

Для перевода канала контроллера в режим широтно-импульсного выхода, необходимо:

- в соответствующий каналу «Регистр конфигурации канала №XX» передать значение константы, соответствующей выбранному режиму работы;
- в соответствующий каналу «Регистр периода канала №XX» передать числовое значение от 0 до 65535, устанавливающее период генерируемых импульсов;
- в соответствующий каналу «Регистр выходного значения канала №XX» передать числовое значение от 0 до 65535, устанавливающее заполнение генерируемых импульсов.

***Примечание:** При использовании режима увеличения разрешения системного таймера период и заполнение измеряется в 0,1 миллисекунды, иначе — в целых в миллисекундах.*

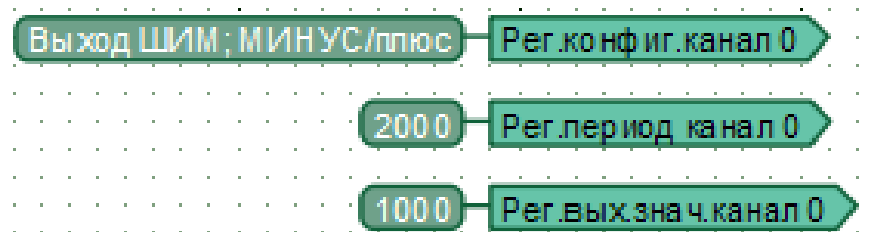


Рисунок 3.15 – Функциональная диаграмма постоянно выполняющая ежесекундное переключение электрического потенциала с +12В(1мА) на GND(120мА) на контакте контроллера соответствующем каналу №0

**ВНИМАНИЕ!** В отличие от CANNY 7, каналы CANNY 5.2 не имеют интеллектуальной защиты от короткого замыкания или перегрузки. Цепи каналов защищены лишь токоограничительными сгораемыми резисторами (см. [схемы](#)). Во избежание выхода каналов из строя, избегайте прямого подключения каналов с включенным выходным потенциалом GND к силовым цепям положительной полярности!

### 3.8.5 Режим дискретного входа

Каналы №№ 2, 8, 9, 10 контроллера могут быть сконфигурированы для работы в режиме дискретного входа. Канал, сконфигурированный для работы в режиме дискретного входа, возвращает значение «1» в регистре своего входного значения, если на соответствующем контакте контроллера установился электрический потенциал соответствующий состоянию «Полярность входа»; и

возвращает значение «0» в регистре входного значения, когда на соответствующем контакте контроллера установился электрический потенциал не соответствующий состоянию «Полярность входа».

Конфигурация канала для работы в данном режиме, определяется константой, задающей полярность электрического потенциала, при появлении которого на соответствующем каналу контакте контроллера в регистр его выходного значения будет записано значение «1».

Изменение потенциала на контактах каналов №№ 2, 8, 9, если их конфигурация отлична от нейтрального состояния, приведет к немедленному автоматическому выходу контроллера из режима пониженного энергопотребления при нахождении контроллера в нём.

Каналы №№ 2, 8, 9, 10 в режиме дискретного входа работают с максимальной чувствительностью, т.е. «защита от дребезга» не предусмотрена и должна быть реализована, при необходимости, в функциональной диаграмме.

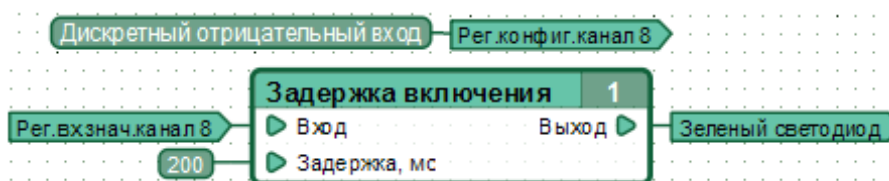


Рисунок 3.16 – Пример функциональной диаграммы, включающей встроенный зеленый светодиод контроллера при поступлении и удержании в течение не менее 200мс на соответствующем каналу №8 контакте контроллера, потенциала GND

### 3.8.6 Режим входа-счетчика

Каналы №№ 2, 8, 9 и 10 контроллера могут быть сконфигурированы для работы в режиме входа-счетчика. Канал сконфигурированный для работы в режиме счетчика, возвращает в регистре своего выходного значения число, соответствующее количеству переключений электрического потенциала на соответствующем контакте контроллера из состояния «Низкий уровень» в состояние «Высокий уровень», или наоборот, за прошедший период времени заданной продолжительности. Таким образом, канал в данном режиме исполняет функцию счетчика передних, либо задних, фронтов сигнала или частотомера, при этом он всегда находится в активном ожидании с максимальной чувствительностью и работает асинхронно функциональной диаграмме, что

позволяет добиться большей точности определения временных параметров исследуемого сигнала и измерять сигнал большей частоты, чем при организации счетчика импульсов средствами функциональной диаграммы.

Конфигурация канала для работы в данном режиме, определяется константой, задающей направление смены полярности входного сигнала, при котором значение счетчика будет увеличено на единицу.

Таблица 3.14 – Параметры конфигурации канала для работы в режиме входа-счетчика

Параметр	Перечень допустимых значений
Тип канала	Дискретный.
Состояние «ВКЛ»	плюс (1мА); МИНУС (120мА).
Состояние «ВЫКЛ»	МИНУС (120мА); плюс (1мА).

Именованные константы, представляющие доступные пользователю комбинации параметров конфигурации каналов, содержатся в разделе «Конфигурация каналов ввода-вывода» справочника констант CannyLab, доступ к которому осуществляется через контекстное меню входа функционального блока, имеющего тип «Константа».

Для перевода канала контроллера в режим входа-счетчика, необходимо:

- в соответствующий каналу «Регистр конфигурации канала №XX» передать значение константы, соответствующей выбранному режиму работы;
- в соответствующий каналу «Регистр периода канала №XX» передать числовое значение от 0 до 65535, устанавливающее период подсчета импульсов в мс.

При ненулевом значении в регистре периода, значение в регистре входного значения канала обновляется один раз в период и содержит число импульсов зарегистрированное счетчиком в течении этого времени.

Подсчет счетчиком импульсов ведется с переполнением. Это означает, что при достижении значения 65535 и последующем увеличении на единицу, значение счетчика устанавливается равным нулю, но каждый последующий импульс вновь увеличивает значение счетчика на единицу.

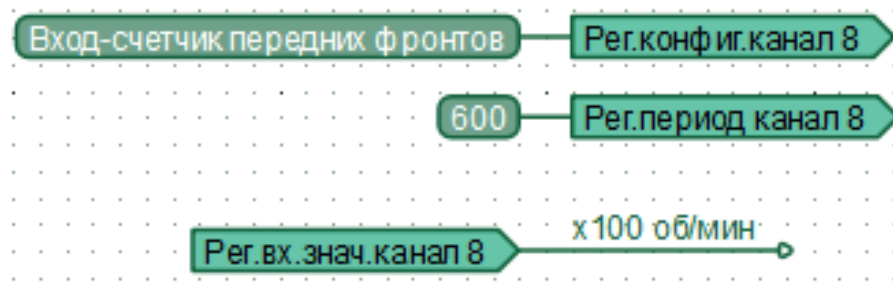


Рисунок 3.17 – Пример функциональной диаграммы тахометра с разрешающей способностью 100 оборотов в минуту, подсчитывающего частоту импульсов на соответствующем канале №8 контакте контроллера.

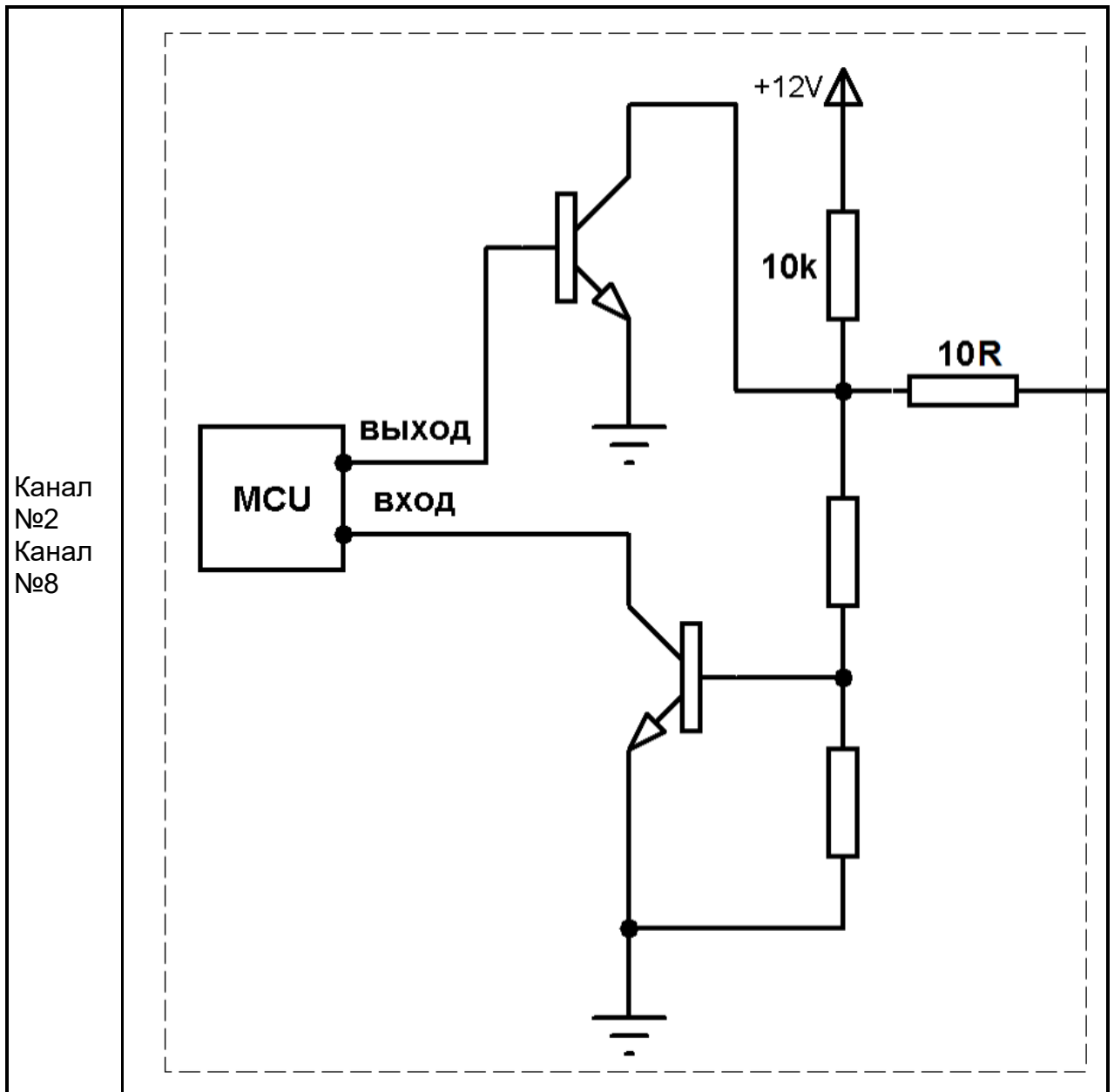
***Примечание:** Если период установлен равным нулю, то счетчик непрерывно ведет подсчет импульсов с накоплением результата в регистре выходного значения канала. В данном режиме значение, этого регистра сбрасывается лишь в результате переполнения регистра.*

### 3.8.7 Эквивалентные принципиальные электрические схемы

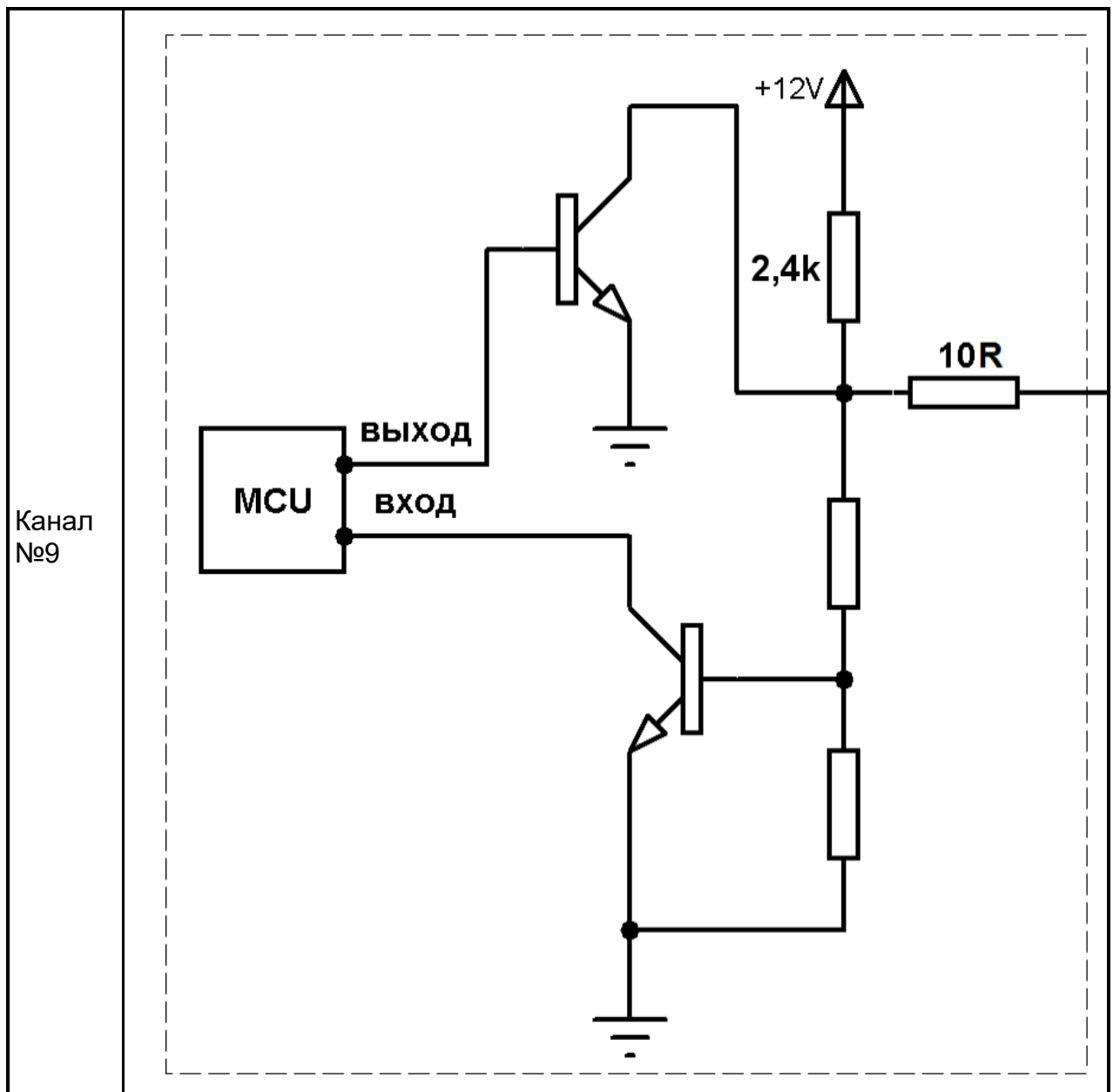
Эквивалентные электрические принципиальные схемы для каждого возможного состояния канала ввода-вывода контроллера приведены в таблице:

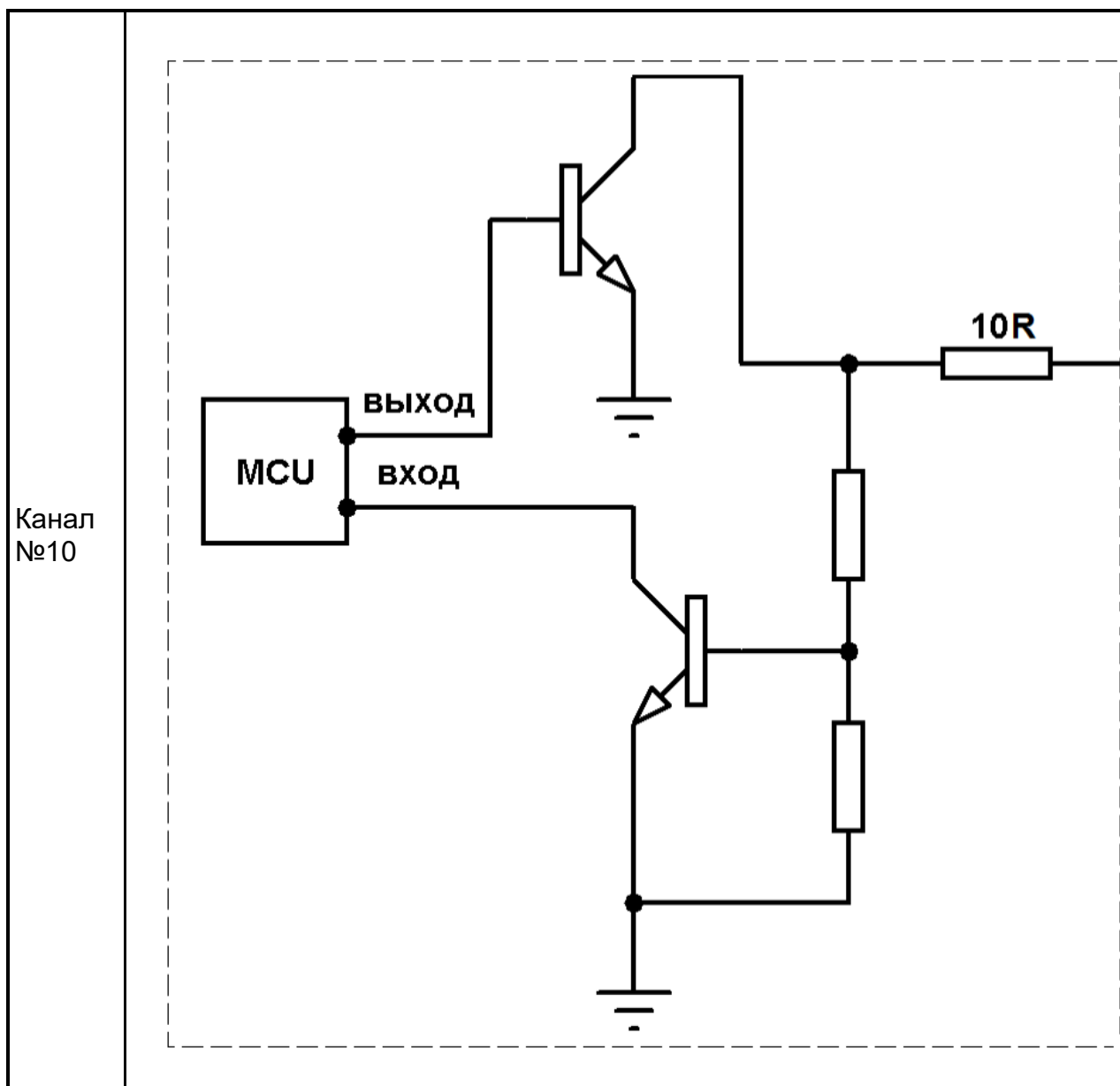
Таблица 3.15 – Эквивалентные электрические принципиальные схемы для состояний канала ввода-вывода контроллера

Регистр	Эквивалентная электрическая принципиальная схема
Канал №0 Канал №1 Канал №3 Канал №4 Канал №5 Канал №6 Канал №7	<p>The diagram illustrates the equivalent electrical circuit for the specified I/O channels. It features a Microcontroller Unit (MCU) connected to the base of a transistor. The collector of this transistor is connected to a +12V power supply through a 10k resistor and a diode. The emitter of the transistor is connected to ground through a diode. A 10R resistor is connected between the collector and the output line.</p>









### 3.9 CANNY 5.2. Драйвер пульта ИК ДУ

- [Общее описание](#)
- [Регистры драйвера пульта ИК ДУ](#)
- [Примеры](#)

#### 3.9.1 Общее описание

Контроллер CANNY 5.2, работая под управлением системного ПО **версии 1.35** и выше, позволяет передавать команды инфракрасных пультов дистанционного управления (ИК ДУ) в широко распространенных форматах NEC и extended NEC. Работа драйвера возможна только на передачу. Только канал №1 контроллера поддерживает работу в режиме передачи данных пультов ИК ДУ.

При передаче команд ИК ДУ, канал №1 контроллера CANNY 5.2 генерирует модулированный высокочастотный (36кГц) сигнал, не требуя для формирования пакетов импульсов наличия источников несущей частоты.

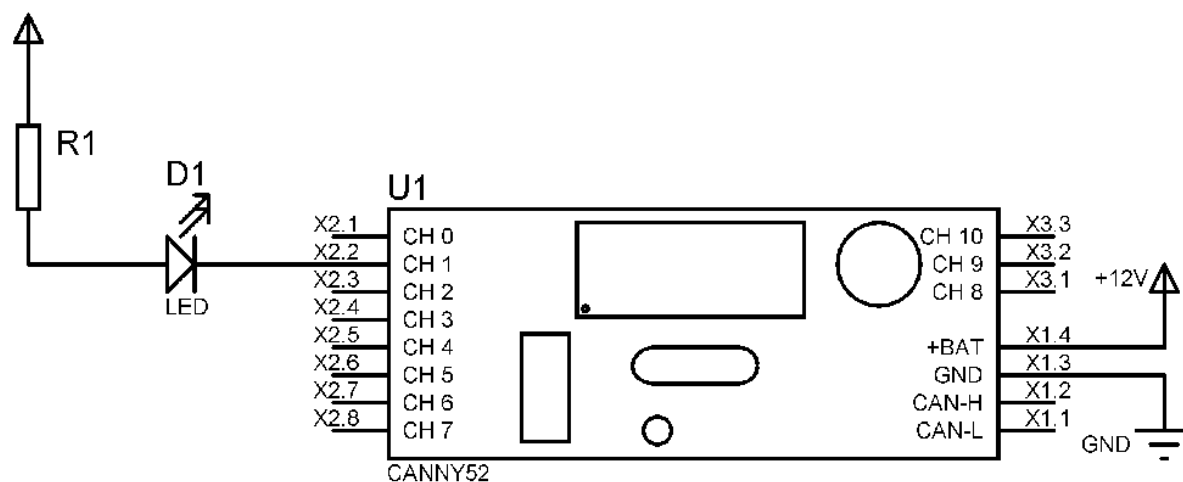


Рисунок 3.18 – Схема подключения ИК ДУ к каналу № 1.

В контроллерах CANNY 5.2, работающих под управлением системного ПО **версии 1.39** и выше, появилась возможность работы канала №1 в режиме передатчика ИК-сигнала без несущей частоты, что позволяет использовать внешние источники несущей с требуемой пользователю частотой.

Драйвер ИК ДУ в своей работе использует ресурсы каналов контроллера, но имеет более высокий приоритет чем драйвер дискретного ввода-вывода. Таким

образом, при активации драйвера ИК ДУ, для задействованном в его работе канале, изменение значений в связанных с ними регистрах драйвера дискретного ввода-вывода будет проигнорировано контроллером.

### 3.9.2 Регистры драйвера пульта ИК ДУ

Драйвер пульта ИК ДУ предусматривает возможность организации передачи данных используя только один канал контроллера (канал №1).

Ниже приведено описание допустимых значений регистров управления работой драйвера пульта ИК ДУ.

Регистры конфигурации драйвера пульта ИК ДУ.

Таблица 3.16 – Допустимые значения регистра конфигурации драйвера пульта ИК ДУ.

Регистр	Ожидаемые значения
Регистр конфигурации ИК-порта	<p><b>1...N</b> = установить конфигурацию драйвера пульта ИК ДУ контроллера, определяющую текущий режим и параметры его работы (задается специальной константой из справочника констант);</p> <p><b>0</b> = отключить драйвер пульта ИК ДУ, вернуть управление каналом драйверу каналов ввода-вывода и разрешить изменения его состояния из функциональной диаграммы.</p>

Конфигурация драйвера задается именованными константами, представляющими комбинацию параметров, определяющих тип сигнала, определяющим электрические потенциалы исходного состояния канала и состояния канала при передаче данных, наличие и потенциал внутренней «подтяжки» канала контроллера.

Таблица 3.17 – Допустимые значения констант конфигурации драйвера пульта ИК ДУ.

Параметр	Перечень допустимых значений
Стандарт	NEC, несущая частота 36кГц; NEC, без несущей
Направление потока данных	Передача



Рисунок 3.19 – Пример конфигурации драйвера пульта ИК ДУ контроллера CANNY 5.2 с несущей частотой.

Таблица 3.18 – Регистры диагностики драйвера пульта ИК ДУ.

Регистр	Возвращаемые значения
Регистр готовности буфера передачи ИК	<b>1</b> = буфер передачи данных драйвера пульта ИК ДУ свободен и готов к работе; <b>0</b> = буфер передачи данных драйвера пульта ИК ДУ занят.
Регистр переполнения буфера драйвера ИК	<b>1</b> = буфер соответствующего канала драйвера пульта ИК ДУ переполнен; <b>0</b> = переполнения соответствующего канала драйвера пульта ИК ДУ не зафиксировано.

Таблица 3.19 – Регистры передачи драйвера пульта ИК ДУ.

Регистр	Ожидаемые значения
Регистр начала передачи ИК	<b>≥ 1</b> = загрузить данные из регистров передачи в буфер передачи драйвера пульта ИК ДУ; <b>0</b> = не загружать данные в буфер ДУ передачи драйвера пульта ИК ДУ.
Регистр сообщения передачи ИК D1:D0 Регистр сообщения передачи ИК D3:D2	<b>0...0xFFFF</b> = значения передаваемых байт сообщения драйвера пульта ИК ДУ по два байта на регистр.

Специальная команда стандарта NEC – «повтор команды», кодируется значениями D1:D0 = 0xFFFF и D3:D2 = 0xFFFF.

*Примечание:* При отправке ИК-сообщений, каждая последующая команда должна передаваться только после освобождения буфера передачи, т. е. при наличии в регистре готовности буфера передачи ИК-сообщения значения «1». Команда повтора предыдущей команды должна отправляться сразу после отправки основной команды, не дожидаясь освобождения буфера передачи, при этом отправка должна выполняться непрерывно, как можно чаще, все время, пока требуется передача подтверждения.

### 3.9.3 Примеры

Пример функциональной диаграммы эмуляции контроллером CANNY 5.2 ИК-пульта ДУ, т.е. передачи ИК-команд управляемому устройству. При появлении

на входе канала №8 значения «1» (нажатия управляющей кнопки), контроллер отправляет ИК-сообщение внешнему устройству. В сообщении содержится адрес 0xAB и команда 0xCD. При длительном сохранении значения «1» на входе канала №10 (удержании кнопки) контроллер отправляет сообщение «повтор команды». Специальная команда стандарта NEC – «повтор команды», кодируется значениями D1:D0 = 0xFFFF и D3:D2 = 0xFFFF.

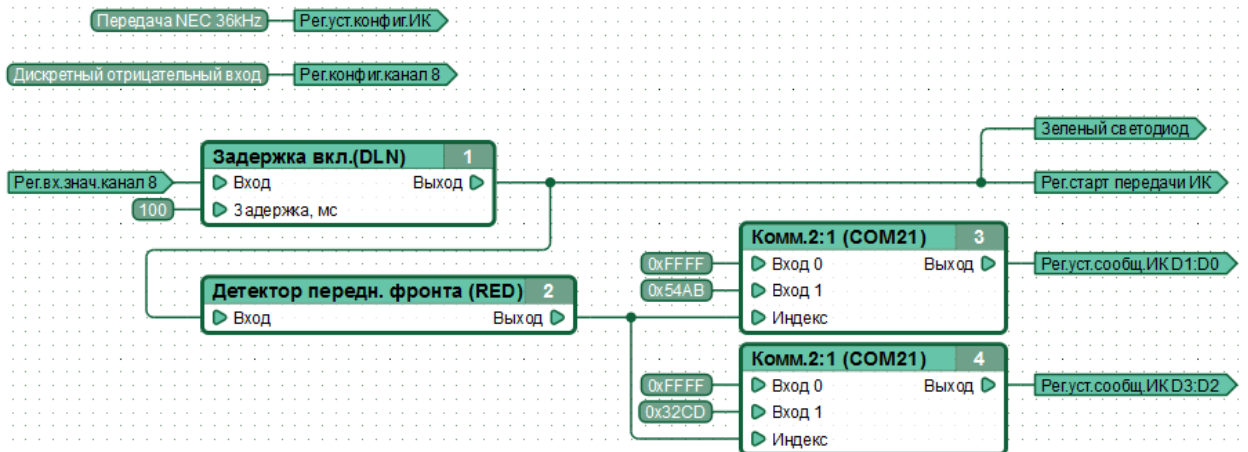


Рисунок 3.20 – Пример функциональной диаграммы эмуляции контроллером CANNY 5.2 ИК-пульта ДУ.

### **3.10 CANNY 5.2. Системные ресурсы и режимы работы**

- [Общее описание](#)
- [Сброс контроллера](#)
- [Встроенный светодиод контроллера](#)
- [Режим пониженного энергопотребления](#)
- [Изменение разрешения системного таймера](#)
- [Фактическое время выполнения функциональной диаграммы](#)
- [Идентификатор устройства](#)
- [Контроль активности интерфейсов контроллера](#)
- [Идентификатор вендора устройства](#)

#### **3.10.1 Общее описание**

Системные ресурсы контроллера отображаются на группу регистров чтения и группу регистров записи. Обращаясь к данным регистрам из функциональной диаграммы, можно получить востребованные в практическом применении сведения о текущем состоянии контроллера и управлять режимами его работы. Список регистров системных ресурсов находится в разделе «Состояние контроллера» справочника регистров, который доступен пользователю через контекстное меню элементов «Регистр чтения» и «Регистр записи».

#### **3.10.2 Сброс контроллера**

Сброс контроллера происходит в результате любого из двух событий: при включении питания контроллера, при программном сбросе из функциональной диаграммы. При сбросе выполняется инициализация контроллера: все содержимое оперативной памяти очищается, каналы ввода-вывода переводятся в нейтральное состояние, драйверы системного программного переводятся в исходное состояние, устанавливается режим нормального энергопотребления и выполнение функциональной диаграммы начинается с начала. Содержимое энергонезависимой памяти контроллера при сбросе не изменяется.

Информация о том, что произошел сброс доступна при обращении к регистру «Регистр контроля восстановления питания».

Таблица 3.20 – Значения возвращаемые регистром контроля восстановления питания.

Регистр	Возвращаемые значения
Регистр контроля восстановления питания	<b>1</b> = текущий цикл выполнения диаграммы является первым с момента программного сброса или восстановления питания контроллера; <b>0</b> = текущий цикл выполнения диаграммы не является первым с момента сброса или восстановления питания.

Принудительный сброс контроллера производится записью ненулевого значения в «Регистр сброса». В этом случае сброс контроллера происходит немедленно после окончания цикла выполнения функциональной диаграммы и установки нового состояния контроллера, в ходе которого произошла такая запись.

Таблица 3.21 – Параметры регистра сброса.

Регистр	Ожидаемые значения
Регистр сброса	$\geq 1$ = запустить процедуру принудительного сброса контроллера; <b>0</b> = значение игнорируется

### 3.10.3 Встроенный светодиод контроллера

Контроллер имеет встроенный зеленый светодиод, управление включением которого осуществляется из функциональной диаграммы путем записи значений в соответствующий регистр.

Таблица 3.22 – Параметры регистра включения зеленого светодиода.

Регистр	Ожидаемые значения
Регистр включения зеленого светодиода	$\geq 1$ = включить встроенный зеленый светодиод контроллера; <b>0</b> = выключить встроенный зеленый светодиод контроллера.

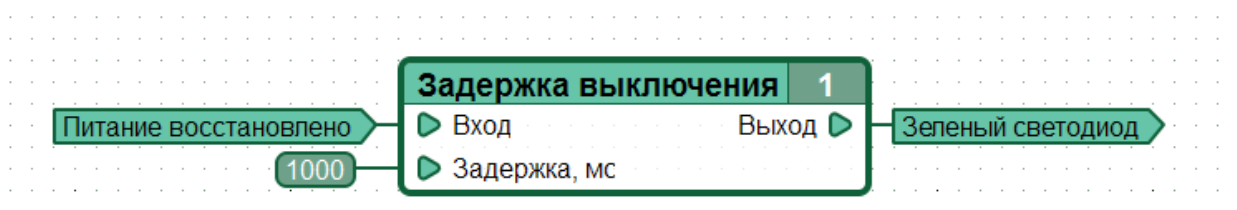


Рисунок 3.21 – Фрагмент функциональной диаграммы, включающий встроенный зеленый светодиод контроллера на одну секунду после каждого сброса контроллера.



### 3.10.4 Режим пониженного энергопотребления

После сброса контроллер начинает работу в режиме нормального энергопотребления, функциональная диаграмма выполняется непрерывно. Переход в режим пониженного энергопотребления осуществляется по команде функциональной диаграммы, записью ненулевого значения в «Регистр установки режима пониженного энергопотребления». Переход в режим пониженного энергопотребления происходит немедленно после окончания цикла выполнения функциональной диаграммы, в ходе которого была произведена такая запись, в отсутствие условий, препятствующих этому переходу.

Продолжительность фазы «сна» составляет 1024 мс. Это означает, что находясь в режиме пониженного энергопотребления, в отсутствие условий перехода в режим нормального энергопотребления, контроллер делает паузу продолжительностью около 1 секунды после каждого цикла выполнения функциональной диаграммы.

Таблица 3.23 – Параметры регистра установки режима пониженного энергопотребления.

Регистр	Ожидаемые значения
Регистр установки режима пониженного энергопотребления	$\geq 1$ = перейти в режим пониженного энергопотребления; <b>0</b> = вернуться в режим нормального энергопотребления.

Возврат контроллера в режим нормального энергопотребления происходит либо принудительно: немедленно после окончания цикла выполнения функциональной диаграммы, в ходе которого было записано значение «0» в «Регистр установки режима пониженного энергопотребления», либо автоматически в результате любого из следующих событий:

- при изменении электрического потенциала на контактах контроллера, соответствующих каналам №№2, 8, 9, 10;
- при включенном из функциональной диаграммы драйвере CAN, при изменении электрического потенциала на соответствующем драйверу контакте контроллера.

**Примечание:** Информация об изменениях электрического потенциала на контактах контроллера, соответствующих каналам №№2, 8, 9, 10, и изменениях электрического потенциала на контактах контроллера, соответствующих включенному драйверу CAN, доступна пользователю через специальный регистр состояния контроллера - «Регистр контроля активности интерфейсов контроллера».

Информация о текущем режиме энергопотребления доступна при обращении к регистру «Регистр контроля режима энергопотребления».

Таблица 3.24 – Значения возвращаемые регистром контроля режима энергопотребления.

Регистр	Возвращаемые значения
Регистр контроля режима энергопотребления	<b>1</b> = контроллер находится в режиме пониженного энергопотребления; <b>0</b> = контроллер находится в режиме нормального энергопотребления

**Примечание:** При создании функциональных диаграмм, использующих режим пониженного энергопотребления, следует учитывать побочный эффект привносимый изменением масштаба времени. Эффект выражается в том, что приращение счетчиков времени функциональных блоков: задержек включения, выключения и генераторов ШИМ в режиме пониженного энергопотребления происходит скачкообразно, в соответствии с временем фактически проведенным в фазе «сна» (1024 мс).

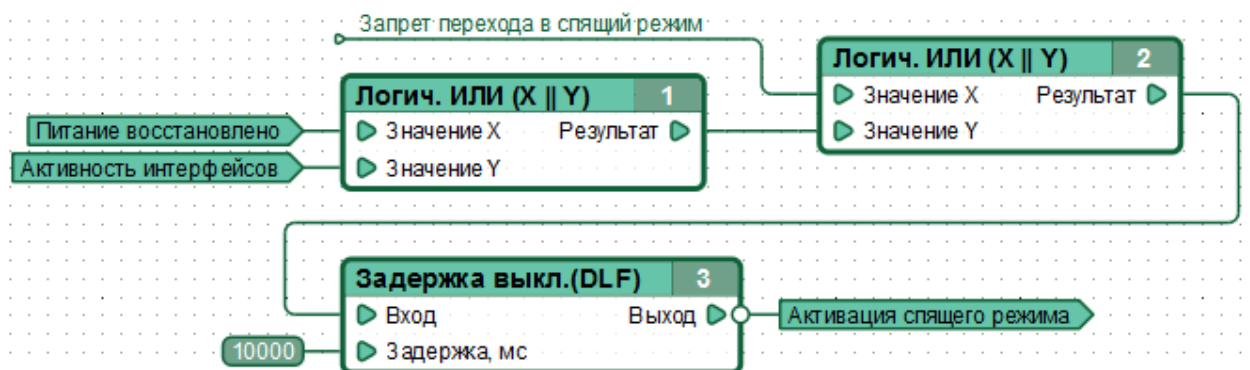


Рисунок 3.22 – Фрагмент функциональной диаграммы, реализующий типовое управление режимом пониженного энергопотребления: переход в режим пониженного энергопотребления в отсутствие в течение 10 секунд условий препятствующих этому и автоматический возврат в нормальный режим при

активности периферии контроллера или по установке запрета «засыпания» из диаграммы.

*Примечание: Обратите внимание на инверсию по выходу функционального блока №3.*

### 3.10.5 Изменение разрешения системного таймера

Для расширения возможностей контроллера при работе с временными интервалами, например в качестве генератора импульсов или счетчика, предусмотрен режим увеличения разрешения системного таймера. По умолчанию этот режим отключен, а управление им осуществляется из функциональной диаграммы, путем записи ненулевого значения в соответствующий регистр контроллера.

Таблица 3.25 – Параметры регистра увеличения разрешения системного таймера x10.

Регистр	Ожидаемые значения
Регистр увеличения разрешения системного таймера x10	$\geq 1$ = установить разрешение системного таймера 0,1мс (100мкс); <b>0</b> = установить разрешение системного таймера 1мс.

Если было включено увеличение разрешения системного таймера, то все значения временных интервалов, используемых в функциональной диаграмме и регистрах драйверов, такие как задержка включения, задержка выключения, генератор ШИМ и т. п., начинают измеряться в единицах x0,1мс, т. е. установка интервала времени, равного 100мс, будет соответствовать указанию числа 1000, при вводе соответствующей константы (100мс = 1000 x 0,1мс).

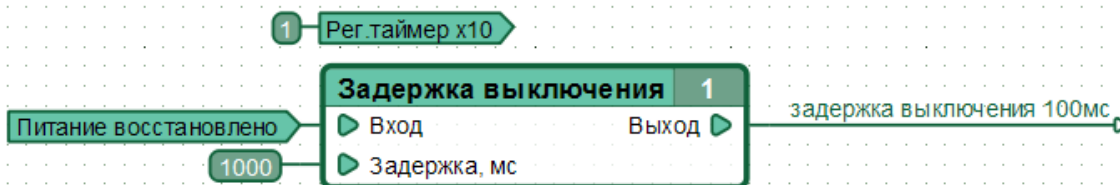


Рисунок 3.23 – Фрагмент функциональной диаграммы, устанавливающей задержку выключения 100мс, при использовании режима увеличения разрешения системного таймера.

### 3.10.6 Фактическое время выполнения функциональной диаграммы

Время, требующееся контроллеру CANNY 5.2 для выполнения функциональной диаграммы в реальных условиях эксплуатации зависит от числа и типов функциональных блоков присутствующих на диаграмме, числа задействованных драйверов входящих в состав системного программного обеспечения и их активности. На практике, цикл выполнения диаграммы содержащей около 100 функциональных блоков и активно взаимодействующей с драйвером CAN продолжается ~30 мс.

*Примечание: При создании функциональной диаграммы, следует учитывать эффект привносимый продолжительностью её цикла. Эффект выражается в том, что приращение счетчиков времени функциональных блоков: задержек включения, выключения и генераторов ШИМ происходит скачкообразно. Так, при фактической длительности цикла в равной 10 мс, период всех генераторов ШИМ на диаграмме будет кратен 10 мс.*

Таблица 3.26 – Значения возвращаемые регистром контроля длительности программного цикла, мс.

Регистр	Возвращаемые значения
Регистр контроля длительности программного цикла, мс	<b>0...65535</b> = продолжительность предыдущего полного цикла выполнения функциональной диаграммы в целых долях миллисекунд.

*Примечание: При включенном режиме увеличения разрешения системного таймера, значение в Регистре контроля длительности программного цикла будет содержать значение не в миллисекундах, а в сотнях микросекунд.*

Наиболее точным способом измерения общего времени работы контроллера, например при реализации часов, является суммирование с накоплением значений получаемых из регистра «Регистр контроля длительности программного цикла» в ходе каждого цикла выполнения функциональной диаграммы.

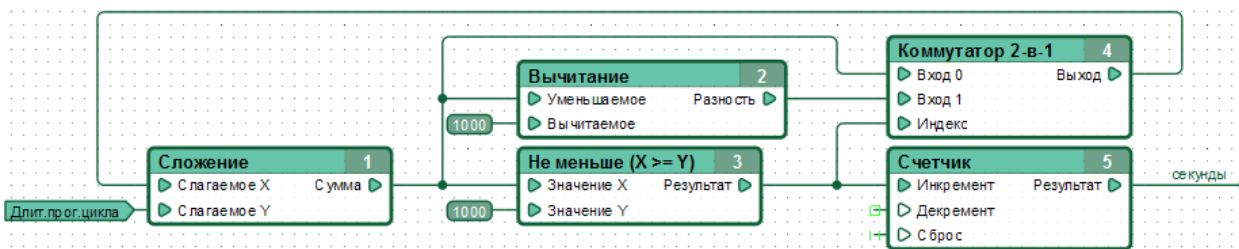


Рисунок 3.24 – Фрагмент функциональной диаграммы, реализующий высокоточный счетчик секунд, пригодный для использования в часах реального времени.

### 3.10.7 Идентификатор устройства

С выходом обновленного системного загрузчика контроллеров CANNY 5.2 версии 001004, при изготовлении устройств, каждому из них присваивается свой идентификационный номер, который можно использовать в дальнейшем при разработке пользовательских диаграмм для дополнительной их защиты от несанкционированного использования.

Доступ к работе с идентификатором устройства осуществляется через соответствующие специальные системные регистры контроллера.

Таблица 3.27 – Значения возвращаемые регистрами идентификаторов устройств.

Регистр	Возвращаемые значения
Регистр идентификатора устройства D1:D0	<b>0...0xFFFF</b> = значение двух младших байт (D1 и D0) индивидуального идентификационного номера контроллера;
Регистр идентификатора устройства D3:D2	<b>0...0xFFFF</b> = значение двух старших байт (D3 и D2) индивидуального идентификационного номера контроллера

В процессе разработки пользовательской диаграммы, из CannyLab, идентификатор устройства можно узнать обратившись к информации об устройстве, доступной в пункте «Устройство» → «Информация» главного меню программы или по нажатию кнопки «Информация» панели инструментов, где он представлен в виде 4х байтового (32-битного) числа, с расположением старшего байта слева.

Например, идентификатор 0x563B8693 будет представлен так: регистр идентификатора устройства D1:D0 равен 0x8693, регистр идентификатора устройства D3:D2 равен 0x563B.

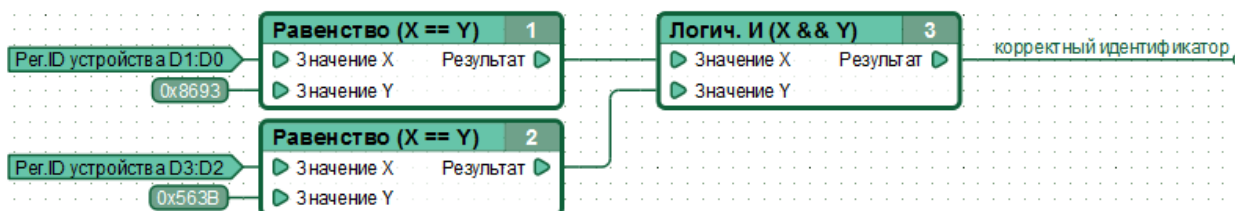


Рисунок 3.25 – Пример функциональной диаграммы, иллюстрирующей работу с идентификатором устройства.

В диаграмме значение, прочитанное из регистров идентификатора устройства, сравнивается с заданными и в случае их совпадения в именованную сеть «корректный идентификатор» сохраняется значение «1».

### 3.10.8 Контроль активности интерфейсов контроллера

«Регистр контроля активности интерфейсов контроллера» - синтетический регистр, отражающий текущую активность задействованных в пользовательской диаграмме внешних интерфейсов контроллера, либо включенных в режиме счетчика или в режиме активного входа каналов ввода-вывода. В те моменты времени, когда по задействованным пользователем интерфейсам контроллера CAN/UART и т.д. не осуществляется прием либо передача каких-либо сигналов и не происходит изменений электрического потенциала на соответствующих активным каналам-входам контактах контроллера, в «Регистре контроля активности интерфейсов контроллера» находится значение «0».

Использование данного регистра удобно в алгоритмах [управления режимами энергопотребления контроллера](#).

Таблица 3.28 – Значения возвращаемые регистром контроля активности интерфейсов контроллера.

Регистр	Возвращаемые значения
Регистр контроля активности интерфейсов контроллера	<p><math>\geq 1</math> = в течении предыдущего цикла выполнения диаграммы, на одном или нескольких задействованных в диаграмме интерфейсах или активных каналах ввода-вывода обнаружена активность</p> <p><math>0</math> = в течении предыдущего цикла выполнения диаграммы, ни на одном задействованном в диаграмме интерфейсе контроллера или активном канале ввода-вывода активности не обнаружено</p>

### 3.10.9 Идентификатор вендора устройства

С выходом обновленного системного загрузчика контроллеров CANNY 5.2 версии 001005, при изготовлении устройств, каждому из них присваивается идентификационный номер их вендора (поставщика), который можно использовать в дальнейшем при разработке пользовательских диаграмм для дополнительной их защиты от несанкционированного использования.

Идентификатор вендора устройства, назначаемый производителем контроллеров, является одинаковым для всех контроллеров, предназначенных для одного контрагента, или может быть установлен отдельно на конкретную партию контроллеров. Для его производства контроллеров с конкретным идентификатором вендора необходимо обратиться к производителю.

Доступ к работе с идентификатором устройства осуществляется через соответствующий специальный системный регистр контроллера.

Таблица 3.29 – Значения возвращаемые регистром идентификатора вендора устройства.

Регистр	Возвращаемые значения
Регистр идентификатора вендора устройства	0...0xFFFF = значение индивидуального идентификационного номера вендора.

В процессе разработки пользовательской диаграммы, из CannyLab, идентификатор устройства можно узнать обратившись к информации об устройстве, доступной в пункте «Устройство» → «Информация» главного меню программы или по нажатию кнопки «Информация» панели инструментов, где он представлен в виде 2х байтового (16-битного) числа, с расположением старшего байта слева.

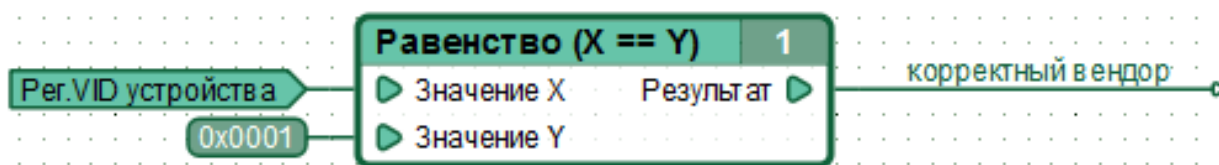


Рисунок 3.26 – Пример функциональной диаграммы, иллюстрирующей работу с идентификатором вендора устройства.

В диаграмме значение, прочитанное из регистра идентификатора вендора устройства, сравнивается с заданными и в случае их совпадения в именованную сеть «корректный вендор» сохраняется значение «1».

### 3.11 CANNY 5.2. Драйвер UART - RS232

- [Общее описание](#)
- [Регистры драйвера](#)
- [Работа контроллера в режиме UART](#)

#### 3.11.1 Общее описание

Контроллеры CANNY 5.2 имеют аппаратный интерфейс **UART** поддерживающий работу в асинхронном дуплексном режиме приема/передачи данных и могут быть использованы для связи контроллеров друг с другом или с внешним оборудованием поддерживающим данный протокол связи. Интерфейс UART1 задействует каналы №9 (TX) и №10 (RX) контроллера. При включении интерфейса активируются оба его канала.

Объем приемного и передающего буферов данных драйвера UART CANNY 5.2 составляет 32 байта.

Набор поддерживаемых скоростей ограничен: 300, 1200, 2400, 9600, 10400, 19200, 57600, 115200 бод; а также доступен только один формат передачи данных: 8-N-1.

Особенности реализации UART - RS232 в контроллере CANNY 5.2:

- Режимы работы: UART, RS-232
- Скорость передачи данных (бод) - 300; 1200; 2400; 9600; 10400; 19200; 57600; 115200
- Формат передачи данных: 8-N-1
- Объем приемного и передающего буферов данных драйвера UART / RS-232 : 32 байта.
- Прием и передача сообщения осуществляется в кодировке **ASCII**.

***Примечание:** Для корректной работы всех протоколов на базе **UART / RS-232** необходимо, чтобы контакты GND устройств, совершающих обмен данными, были приведены к единому потенциалу («общая земля»).*

Драйвер UART / RS232 в своей работе использует ресурсы каналов контроллера, но имеет более высокий приоритет чем драйвер дискретного ввода-



вывода. Таким образом, при активации того или иного интерфейса UART / RS232, для задействованных в его работе каналов, изменение значений в связанных с ними регистрах драйвера дискретного ввода-вывода будет проигнорировано контроллером.

### 3.11.2 Регистры драйвера

Ниже приведено описание допустимых и возвращаемых значений регистров управления работой драйвера.

Таблица 3.30 – Регистры конфигурации драйвера UART.

Регистр	Ожидаемые значения
Регистр конфигурации UART1	<b>1...N</b> = установить конфигурацию канала драйвера UART контроллера, определяющую текущий режим и параметры его работы (задается специальной константой из справочника констант). <b>0</b> = отключить канал от драйвера UART вернуть управление каналом драйверу каналов ввода-вывода и разрешить изменения его состояния из функциональной диаграммы
Регистр установки таймаута приема сообщения UART1, мс	<b>1...65535</b> = прекращение приема данных, если в течении указанного времени на линии не было зафиксировано ни одного изменения и линия находится в пассивном состоянии. <b>0</b> = использовать значение по умолчанию (1мс).
Регистр включения режима открытого коллектора канала UART1 TX	<b>≥ 1</b> = отключить верхнее плечо канала передачи соответствующего интерфейса UART, передача будет в режиме «воздух / GND». <b>0</b> = включить верхнее и нижнее плечо канала передачи соответствующего интерфейса UART передача будет осуществляться в режиме «+5В / GND».

Конфигурация драйвера UART определяется константой, представляющей комбинацию параметров, определяющих скорость, режим, дополнительные параметры передачи данных и потенциал линии в пассивном режиме.

Таблица 3.31 – Параметры конфигурации драйвера UART.

Параметр	Перечень допустимых значений
Скорость передачи данных, бод	300, 1200, 2400, 9600, 10400; 19200, 57600, 115200
Режим работы	UART 8-N-1
Полярность	прямая (UART); инверсная (RS-232)

Режим открытого коллектора	«ВКЛ»; «ВЫКЛ»
----------------------------	---------------

Именованные константы, представляющие доступные пользователю комбинации параметров конфигурации UART, содержатся в разделе «Конфигурация UART» справочника констант CannyLab, доступ к которому осуществляется через контекстное меню констант на функциональной диаграмме.

Таблица 3.32 – Регистры диагностики драйвера UART.

Регистр	Возвращаемые значения
Регистр переполнения буфера UART1	<b>1</b> = буфер UART переполнен; <b>0</b> = переполнения не зафиксировано.
Регистр отсутствия активности драйвера UART1	<b>1</b> = активность соответствующего канала драйвера UART отсутствует, линия находится в пассивном режиме; <b>0</b> = зафиксирована активность на линии соответствующего канала драйвера UART.
Регистр ошибки приема UART1	<b>1</b> = во время приема данных UART произошла ошибка; <b>0</b> = драйвер работает в нормальном режиме.
Регистр готовности буфера буфера передачи данных UART1	<b>1</b> = буфер передачи данных драйвера UART свободен; <b>0</b> = буфер передачи данных драйвера UART занят, передача данных невозможна.

Таблица 3.33 – Регистры приема драйвера UART.

Регистр	Возвращаемые значения
Регистр наличия принятых данных UART1	<b>1</b> = сообщение получено и помещено; в буфер приема соответствующего канала драйвера UART; <b>0</b> = в буфере приема соответствующего канала драйвера UART отсутствуют актуальные данные.
Регистр длины принятого сообщения UART1	<b>0...32</b> = значение, равное количеству байт данных, в принятом по соответствующему каналу драйвера UART пакете данных.
Регистр принятого сообщения UART1 D1:D0 ... Регистр принятого сообщения UART1 D31:D30	<b>0...0xFFFF</b> = значения соответствующих байт данных приемных буферов UART каждого канала, по два байта на регистр.

Таблица 3.34 – Регистры передачи драйвера UART.

Регистр	Ожидаемые значения
Регистр начала передачи UART1	$\geq 1$ = загрузить данные из регистров передачи в буфер передачи соответствующего канала драйвера UART; 0 = не загружать данные в буфер передачи соответствующего канала драйвера UART.
Регистр длины сообщения передачи UART1	0...32 = количество байт данных, которое будет необходимо передать в линию, при получении команды на отправку соответствующего канала драйвера UART.
Регистр сообщения передачи UART1 D1:D0 ... Регистр сообщения передачи UART1 D31:D30	0...0xFFFF = значения соответствующих байт данных для передачи по соответствующему каналу драйвера UART, по два байта на регистр.

### 3.11.3 Работа контроллера в режиме UART

Работая в режиме UART контроллер может осуществлять дуплексный прием/передачу данных.

Пример функциональной диаграммы для обмена данными по UART. Контроллер, получив сообщение длиной 2 байта по интерфейсу UART1, сохраняет их в соответствующей именованной сети для дальнейшей обработки и на следующем цикле выполнения диаграммы отправляет обратно заранее подготовленные 2 байта данных.

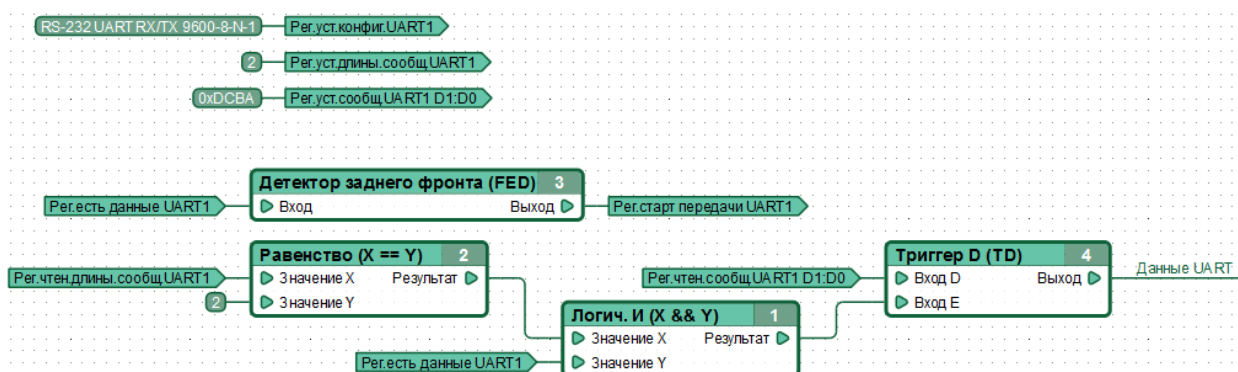


Рисунок 3.27 – Диаграмма подключения

### 3.12 CANNY 5.2. Параметры пользовательской конфигурации

- [Общее описание](#)
- [Регистры параметров пользовательской конфигурации](#)
- [Примеры](#)

#### 3.12.1 Общее описание

Параметры пользовательской конфигурации могут быть заданы конечным пользователем контроллера в момент загрузки в него программного обеспечения с использованием [Исполняемого файла автономной загрузки ПО в контроллер](#). После загрузки ПО и запуска контроллера в автономном режиме, установленные пользователем таким образом данные, становятся доступны функциональной диаграмме в соответствующих регистрах контроллера.

Грамотное использование пользовательских параметров существенно повышает гибкость и универсальность решений на базе контроллера, позволяя конечному пользователю, не имеющему навыков работы с CannyLab, вносить безопасные изменения в работу алгоритма контроллера используя простой пользовательский интерфейс.

#### 3.12.2 Регистры параметров пользовательской конфигурации

Возможно задать до 16 пользовательских параметров, которые будут доступны в 16 соответствующих регистрах контроллера.

Таблица 3.35 – Допустимые значения регистров параметров пользовательской конфигурации.

Регистр	Ожидаемые значения
Регистр параметра пользовательской конфигурации №0 ... Регистр параметра пользовательской конфигурации №15	<b>0...65535</b> = значение соответствующего пользовательского параметра.

Значения в данных регистрах установятся при старте контроллера, после записи в него ПО посредством исполняемого файла автономной загрузки, и будут оставаться неизменными (константными) на протяжении всего времени работы функциональной диаграммы, не изменяясь даже при сбросе контроллера.

Изменить значения данных регистров можно лишь стерев или перезаписав память контроллера новым ПО.

Значения регистров соответствующих параметрам не перечисленным в исполняемом файле автономной устанавливается равным нулю.

***Примечание:** При записи контроллера из среды CannyLab значения всех регистров параметров пользовательской конфигурации устанавливается равным нулю.*

### 3.12.3 Пример использования параметров пользовательской конфигурации

Создавая в среде CannyLab исполняемый файл автономной загрузки ПО в контроллер, указав файл системного ПО контроллера и файл, содержащий приведенную ниже диаграмму, задайте два пользовательский параметра: Имя «Длительность выходного сигнала,мс», Тип «Число» и Имя «Выходной канал контроллера», Тип «Список». В список значений параметра «Выходной канал контроллера» добавьте две строки: Название «Канал 0», Значение «0» и «Канал 1», Значение «1».

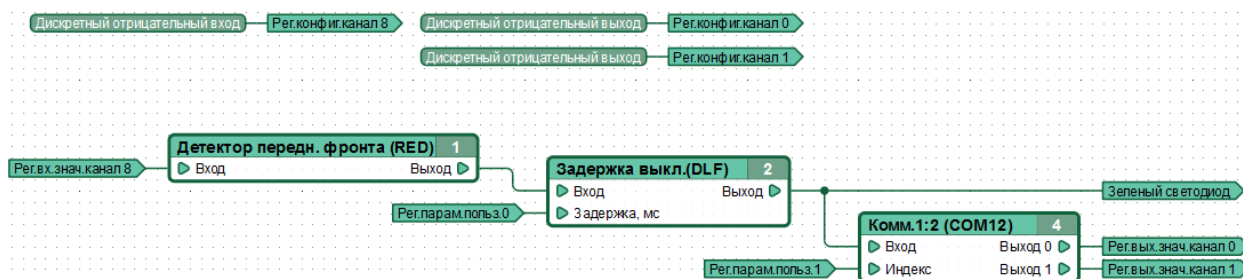


Рисунок 3.28 – Пример диаграммы для использования параметров пользовательской конфигурации.

Запустите созданный таким образом исполняемый файла автономной загрузки ПО, установите требуемые значения параметров и запишите ПО в контроллер.

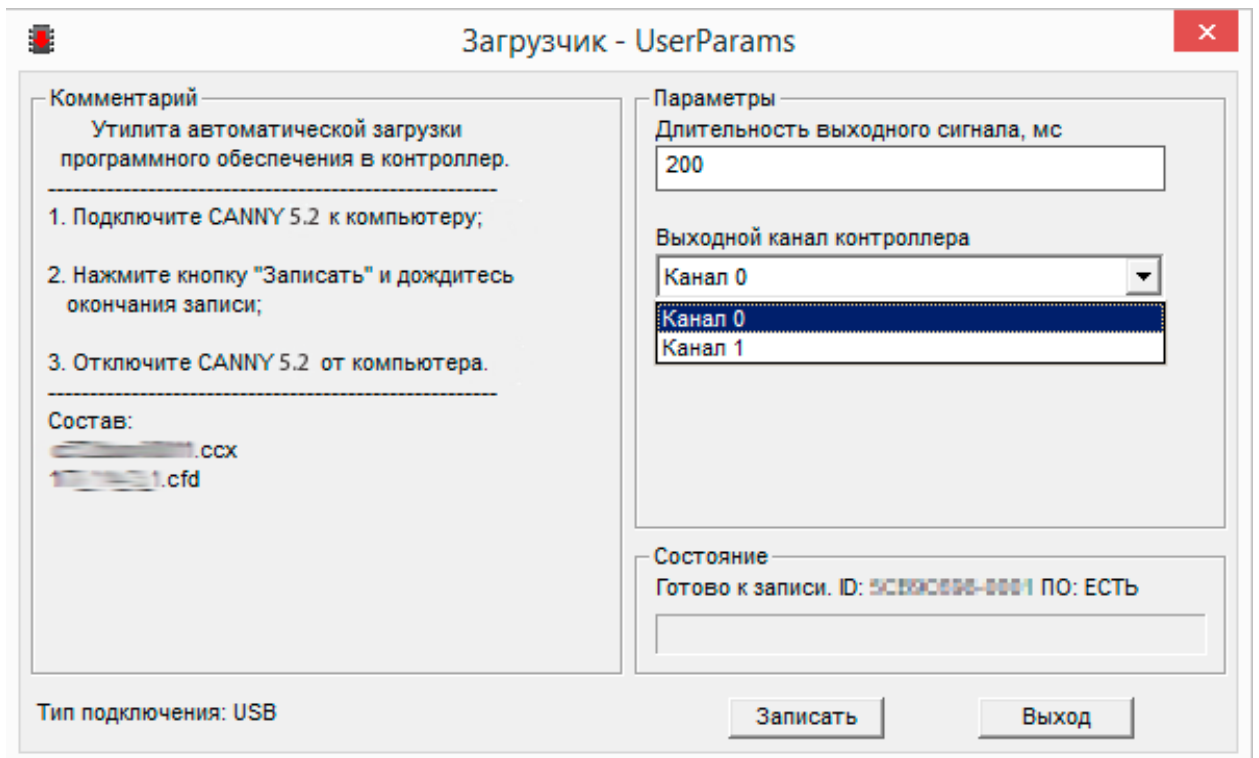


Рисунок 3.29 – Окно программы загрузчика Canny X.

Выполняя диаграмму контроллер, в момент получения на входе канала №8 отрицательного потенциала, устанавливает на заданном пользователем в соответствующем параметре канале потенциал «GND» и удерживает его заданное пользователем время. Для наглядности, в диаграмме реализована индикация состояния выходного канала контрольным светодиодом.

## **4 Заключение**

В случае если у вас остались вопросы, рекомендуем посетить [наш форум](#).

Также информации материалы о продукции представлены на нашем [YouTube канале](#)