

# **Избранные статьи из руководства по программированию и применению контроллеров CANNY®**

**Данный документ автоматически сформирован на  
основании содержания online-энциклопедии CANNY.**

**Актуальная версия документации доступна на сайте  
<http://wiki.canny.ru/>**

<b>1 CANNY 5.2 duo</b>	<b>1</b>
1.1 Общие сведения	1
1.2 Устройство и принцип работы	2
1.3 Режимы работы	5
1.4 Среда исполнения функциональных диаграмм	7
1.5 Ресурсы контроллера	9
<b>2 CANNY 5.2 duo, Системные ресурсы и режимы работы</b>	<b>12</b>
2.1 Общее описание	12
2.2 Сброс контроллера	12
2.3 Встроенный светодиод контроллера	13
2.4 Режим пониженного энергопотребления	13
2.5 Изменение разрешения системного таймера	15
2.6 Фактическое время выполнения функциональной диаграммы	16
2.7 Идентификатор устройства	17
2.8 Контроль активности интерфейсов контроллера	18
2.9 Контроль подключения по интерфейсу USB	18
<b>3 CANNY 5.2 duo, Драйвер каналов ввода-вывода</b>	<b>20</b>
3.1 Общее описание	20
3.2 Регистры драйвера	20
3.3 Состояние канала в момент запуска контроллера	22
3.4 Работа с каналами из пользовательской диаграммы	22
3.5 Эквивалентные принципиальные электрические схемы	23
<b>4 CANNY 5.2 duo, Драйвер CAN</b>	<b>25</b>
4.1 Общее описание	25
4.2 Регистры драйвера	25
<b>5 CANNY 5.2 duo, Драйвер шлюза CAN</b>	<b>31</b>
5.1 Общее описание	31
5.2 Регистры драйвера	32
5.3 Блок-схема работы контроллера в режиме CAN-шлюза	36
<b>6 CANNY 5.2 duo, Драйвер UART</b>	<b>39</b>
6.1 Общее описание	39
6.2 Регистры драйвера	39
6.3 Работа контроллера в режиме UART	42

<b>7 CANNY 5.2 duo, Параметры пользовательской конфигурации.....</b>	<b>43</b>
7.1 Общее описание.....	43
7.2 Регистры параметров пользовательской конфигурации.....	43
7.3 Пример использования параметров пользовательской конфигурации.....	44
<b>8 CANNY 5.2 duo, Энергонезависимая память (ЭНП).....</b>	<b>45</b>
8.1 Общее описание.....	45
8.2 Регистры энергонезависимой памяти.....	45

# 1 CANNY 5.2 duo

**CANNY 5.2 duo** — компактный программируемый логический контроллер с двумя интерфейсами CAN, ориентированный на автомобильное, бытовое и промышленное применение.

## 1.1 Общие сведения

Также как и контроллеры CANNY 7, CANNY 5, CANNY 5.2 и CANNY 5 duo, контроллер CANNY 5.2 duo может быть отнесен к классу интеллектуальных реле или NanoPLC.

В отличие от контроллеров CANNY 7, CANNY 5 и CANNY 5.2, контроллер CANNY 5.2 duo имеет 2 интерфейса CAN. Три канала ввода-вывода контроллера способны работать как в режиме выхода, так и в режиме входа, а один из каналов может работать в режиме Аналого-Цифрового преобразователя (АЦП).

Отличием CANNY 5.2 duo от его предшественника, CANNY 5 duo, является поддержка USB интерфейса, который дает возможность загрузки в контроллер пользовательских программ без использования специального программатора - достаточно иметь только кабель-переходник USB-A--miniUSB, а также поддержка интерфейса UART, который через интерфейс USB (USB Virtual COM-port) позволяет обмениваться данными с другими устройствами, например с ПК, в процессе выполнения пользовательской функциональной диаграммы.

К основным особенностям CANNY 5.2 duo можно отнести:

- номинальное напряжение питания 9 / 18В;
- номинальное напряжение каналов ввода-вывода 0 / 12В (18В max);
- 3 канала ввода/вывода отрицательной полярности с максимальным током каждого из них 120мА, достаточного для управления типовыми автомобильными реле;
- два интерфейса CAN 2.0В совместимых с ISO-11898, SAE J2411 широко применяемым в автомобилях;
- один АЦП, поверх одного из каналов ввода-вывода;
- один интерфейс UART через USB Virtual COM-port;
- 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 пользовательскому приложению, способные сохранить критически важные данные при сбоях питания;
- широкий диапазон рабочих температур от -40 до + 85 °С;

- 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 duo используется тот же самый графический язык программирования CFD, что применяется для программирования других контроллеров CANNY и та же среда разработки - CANNY Lab.

Для записи программного обеспечения в контроллер не требуется каких-либо специальных программаторов, загрузка программного обеспечения осуществляется с помощью обычного кабеля-переходника USB-A--miniUSB.

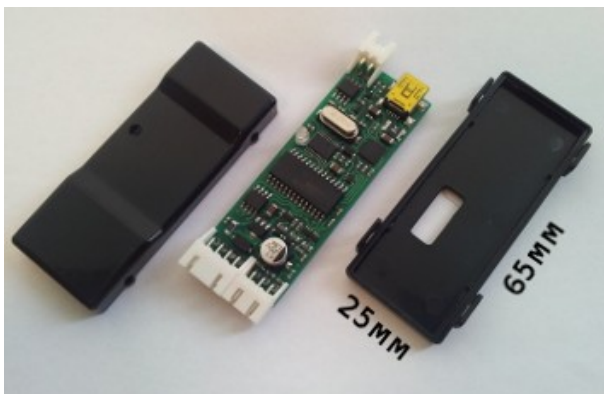
Доступный пользователю объем памяти контроллера способен вместить программы, состоящие из нескольких сотен функциональных блоков, что позволяет реализовать достаточно сложные алгоритмы.

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

## 1.2 Устройство и принцип работы

### 1.2.1 Внешний вид и расположение элементов

Основными конструктивными элементами CANNY 5.2 duo 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.



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

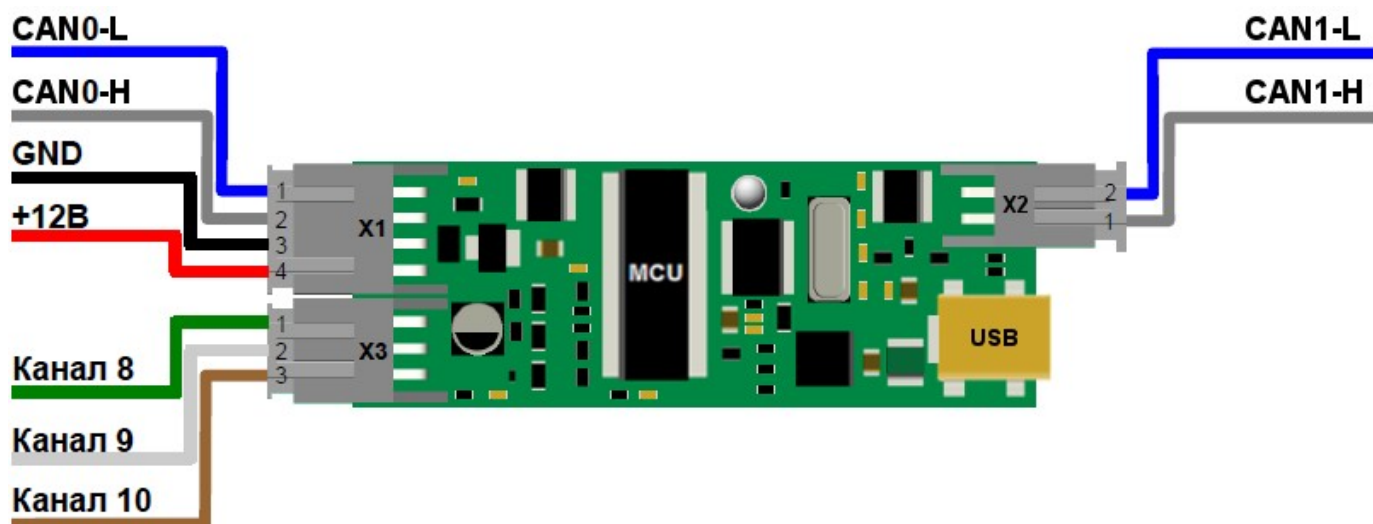
Кабель-переходник USB-A--miniUSB в комплект поставки не входит.

Наружный разъем X1 содержит четыре контакта: вход питания +12В, вход питания GND, CAN0-H и CAN0-L.

Наружный разъем X2 содержит два контакта: CAN1-H и CAN1-L.

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

Наружный разъем USB предназначен для подключения CANNY 5.2 duo к персональному компьютеру и загрузки программного обеспечения в контроллер. Кроме того, USB разъем может использоваться для обмена данными с ПК в процессе выполнения пользовательской диаграммы через USB Virtual COM-port.

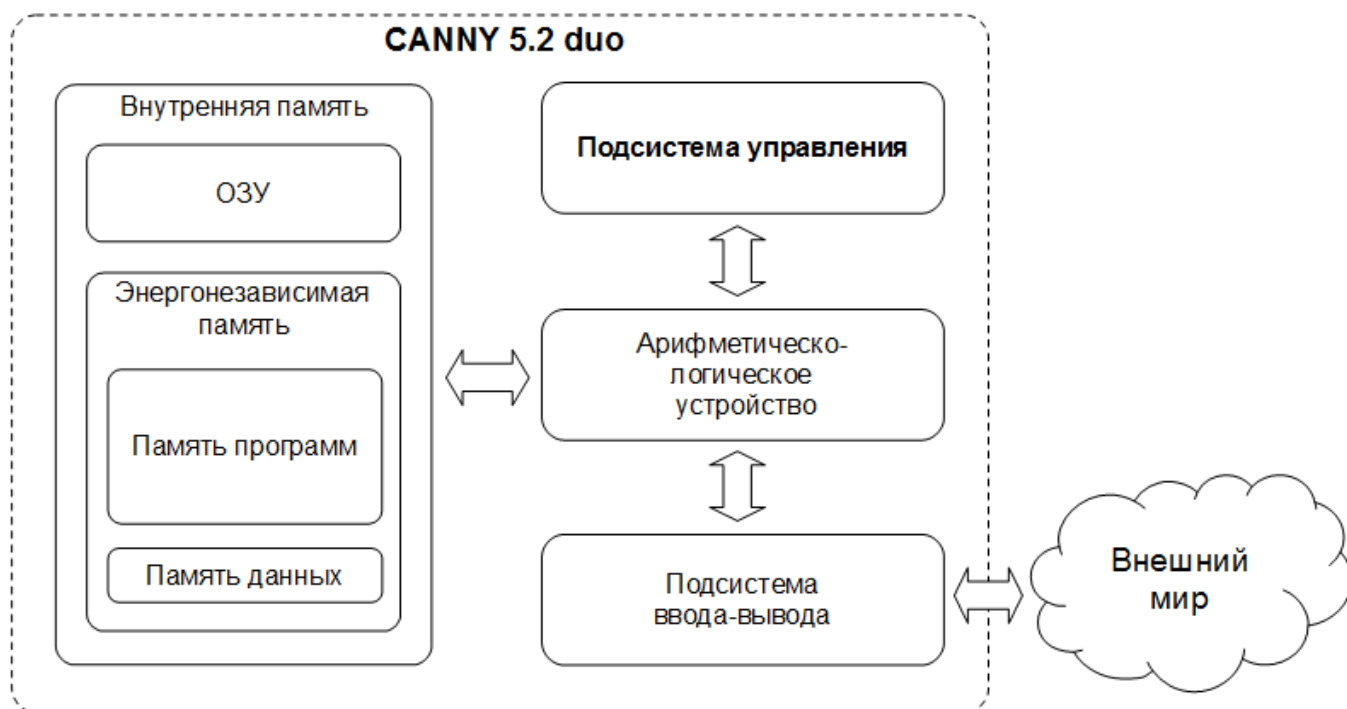


### 1.2.2 Программная архитектура

CANNY 5.2 duo является цифровым программируемым вычислительным управляющим устройством.

В целом, для CANNY 5.2 duo справедливы общие сведения о программируемых логических контроллерах изложенные в статье программируемый логический контроллер.

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



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

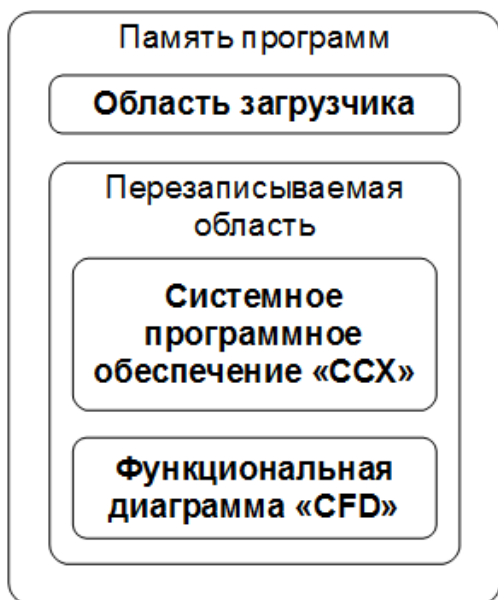
Внутренняя память контроллера разделяется на энергонезависимую память программ, энергонезависимую память данных и оперативную память данных.

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.

### 1.2.3 Структура программного обеспечения

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



Программный загрузчик обеспечивает работу контроллера в режиме загрузки ПО, обеспечивая передачу данных между CANNY 5.2 duo и персональным компьютером, 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 duo распространяется производителем в виде файлов формата СХ 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 СХ 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.

## 1.3 Режимы работы

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



### **1.3.1 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 SannyLab. 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.

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.

### **1.3.2 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.

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

### **1.3.3 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.

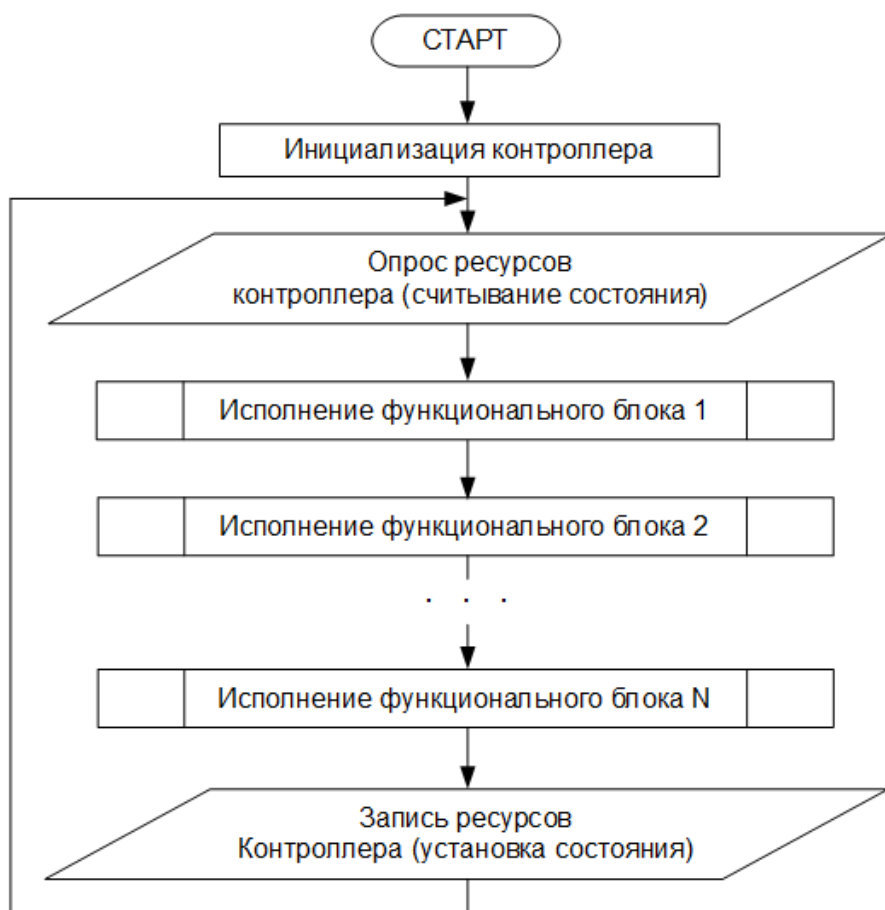
## **1.4 Среда исполнения функциональных диаграмм**

### **1.4.1 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 duo.

### **1.4.2 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*.



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.

### 1.4.3 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.

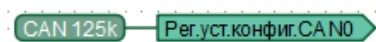
Пользователь имеет возможность указать регистр чтения в качестве источника входных данных практически любого функционального блока на диаграмме и, тем самым, извлечь и использовать при реализации собственных алгоритмов сведения, полученные контроллером из внешнего мира. Например информацию об электрическом 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 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 1 в регистр, расположенный по адресу 1027 задает настройки драйвера CAN0 для работы с шиной на скорости 125кБод.

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 «CAN 125» в регистр с именем «Регистр установки конфигурации CAN0».



Аналогичным образом, по появлению значения «1» в регистре расположенном по адресу 1184 («Регистр входного значения канала №8»), мы можем узнать о приложении отрицательного электрического потенциала к контакту №1 разъема X3 контроллера.



## 1.5 Ресурсы контроллера

### 1.5.1 Системные ресурсы и режимы работы

*Основная статья: CANNY 5.2 duo, Системные ресурсы и режимы работы*

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».

## 1.5.2 Драйвер каналов ввода-вывода

*Основная статья:* **CANNY 5.2 duo, Драйвер каналов ввода-вывода**

Пользователям CANNY 5.2 duo доступны три дискретных канала ввода-вывода общего назначения. Каждый канал физически представлен соответствующим контактом 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 слаботочные цепи управления или, при использовании внешнего защитного диода, небольшие электромагнитные реле. 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.

## 1.5.3 Драйвер CAN

*Основная статья:* **CANNY 5.2 duo, Драйвер CAN**

Контроллеры CANNY 5.2 duo имеют два независимых интерфейса CAN. Два специальных контакта разъема X1, используемые драйвером интерфейса CAN0 (CAN0-H и CAN0-L), и контакты разъема X2, используемые драйвером интерфейса CAN1 (CAN1-H и CAN1-L), предназначены для подключения к цифровой информационной **шине CAN**.

## 1.5.4 Драйвер шлюза CAN

*Основная статья:* **CANNY 5.2 duo, Драйвер шлюза CAN**

Используя драйвер шлюза CAN возможно организовать высокоскоростную асинхронную ретрансляцию сообщений между аппаратными CAN-интерфейсами контроллера, с возможностью автоматической модификации ретранслируемых сообщений по динамически задаваемым диаграммой правилам.

## 1.5.5 Драйвер UART (USB Virtual COM-port)

*Основная статья:* **CANNY 5.2 duo, Драйвер UART**

Драйвер UART, через USB Virtual COM-port, позволяет контроллеру обмениваться данными с внешними устройствами, например ПК, в процессе выполнения пользовательской диаграммы.

## 1.5.6 Параметры пользовательской конфигурации

*Основная статья:* **CANNY 5.2 duo, Параметры пользовательской конфигурации**

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.

## 1.5.7 Энергонезависимая память (ЭНП)

*Основная статья:* **CANNY 5.2 duo, Энергонезависимая память (ЭНП)**

Для предотвращения потери критически важной информации о состоянии пользовательской диаграммы (настройки, коды, текущие режимы работы и т. п.) при сбросе питания либо рестарте, в CANNY 5.2 duo предусмотрен доступ на чтение и запись из пользовательской диаграммы ко встроенной энергонезависимой памяти контроллера.

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

# 2 CANNY 5.2 duo, Системные ресурсы и режимы работы

## 2.1 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».

## 2.2 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».

Регистр	Возвращаемые значения	
Регистр контроля восстановления питания	1	= текущий цикл выполнения диаграммы является первым с момента программного сброса или восстановления питания контроллера
	0	= 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.

Регистр	Ожидаемые значения	
Регистр сброса	$\geq 1$	= запустить процедуру принудительного сброса контроллера
	0	= значение игнорируется

## 2.3 Встроенный светодиод контроллера

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

Регистр	Ожидаемые значения	
Регистр включения зеленого светодиода	$\geq 1$	= включить встроенный зеленый светодиод контроллера
	0	= выключить встроенный зеленый светодиод контроллера
Регистр включения красного светодиода	$\geq 1$	= включить встроенный красный светодиод контроллера
	0	= выключить встроенный красный светодиод контроллера

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



## 2.4 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.

Регистр	Ожидаемые значения	
Регистр режима пониженного энергопотребления	$\geq 1$	= перейти в режим пониженного энергопотребления
	0	= 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 №№8, 9, 10;
- при включенном из функциональной диаграммы драйвере CAN, при изменении электрического потенциала на соответствующих драйверу контактах контроллера.

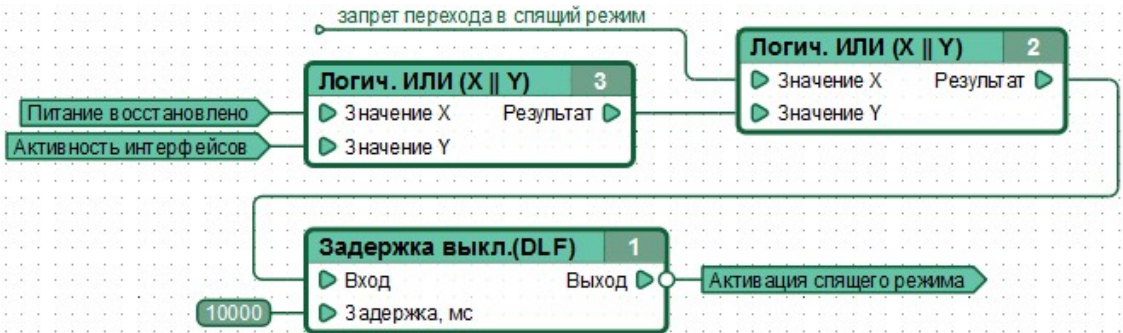
*Примечание: Информация об изменениях электрического потенциала на контактах контроллера, соответствующих каналам №№8, 9, 10, и изменениях электрического потенциала на контактах контроллера, соответствующих включенному драйверу CAN, доступна пользователю через специальный регистр состояния контроллера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

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

Регистр	Возвращаемые значения	
Регистр контроля режима энергопотребления	1	= контроллер находится в режиме пониженного энергопотребления
	0	= 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).*

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.

## 2.5 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.

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

Фрагмент функциональной диаграммы, устанавливающей задержку выключения 100мс, при использовании режима увеличения разрешения системного таймера:



## 2.6 Фактическое время выполнения функциональной диаграммы

Время, требующееся контроллеру CANNY 5.2 duo 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 ~25 мс.

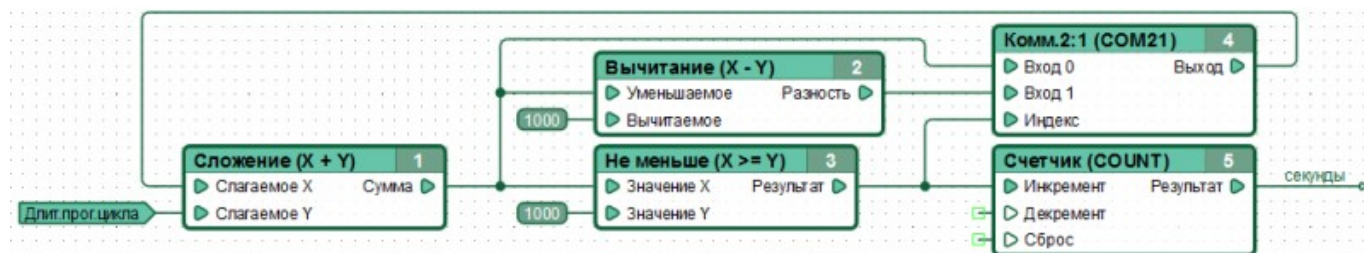
*Примечание: При создании функциональной диаграммы, следует учитывать эффект привносимый продолжительностью её цикла. Эффект выражается в том, что приращение счетчиков времени функциональных блоков: задержек включения, выключения и генераторов ШИМ происходит скачкообразно. Так, при фактической длительности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

Регистр	Возвращаемые значения
Регистр контроля длительности программного цикла, мс	0...65535 = 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.

Фрагмент функциональной диаграммы, реализующий высокоточный счетчик секунд, пригодный для использования в часах реального времени:



## 2.7 Идентификатор устройства

При изготовлении контроллеров CANNY 5.2 duo, каждому из них присваивается свой идентификационный номер, который можно использовать в дальнейшем при разработке пользовательских диаграмм для дополнительной их защиты от несанкционированного использования.

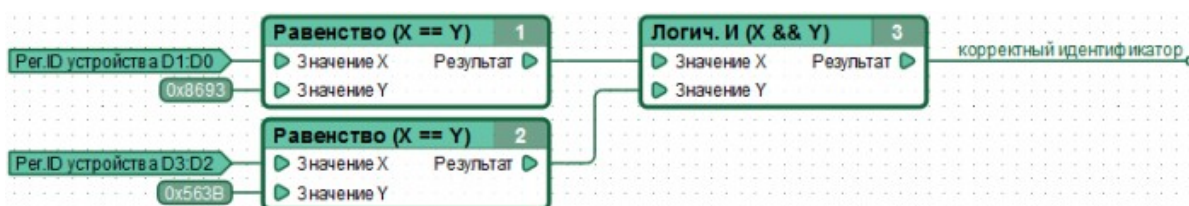
Доступ к работе с идентификатором устройства осуществляется через соответствующие специальные системные регистры контроллера.

Регистр	Возвращаемые значения
Регистр идентификатора устройства D1:D0	0...65535 = значение двух младших байт (D1 и D0) индивидуального идентификационного номера контроллера;
Регистр идентификатора устройства D3:D2	0...65535 = 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.

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».



## 2.8 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 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.

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

## 2.9 Контроль подключения по интерфейсу USB

*"Регистр наличия подключения USB"* - регистр, позволяющий отслеживать статус подключения контроллера к внешним устройствам, например к ПК, с помощью USB-интерфейса в процессе выполнения пользовательской диаграммы, что дает возможность организовывать с ними обмен данными, используя драйвер UART CANNY 5.2 duo.

Регистр	Возвращаемые значения	
Регистр наличия подключения USB	$\geq 1$	= обнаружено подключение контроллера к USB-интерфейсу внешнего устройства
	0	= подключения контроллера к USB-интерфейсам внешних устройств не обнаружено



# 3 CANNY 5.2 duo, Драйвер каналов

## ВВОДА-ВЫВОДА

### 3.1 Общее описание

Пользователям CANNY 5.2 duo доступны три дискретных канала ввода-вывода общего назначения. Каждый канал физически представлен соответствующим контактом 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 слаботочные цепи управления или, при использовании внешнего защитного диода, небольшие электромагнитные реле. 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.

**ВНИМАНИЕ! В отличие от CANNY 7, каналы CANNY 5.2 duo 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!**

### 3.2 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.

Регистр	Ожидаемые значения
Регистр выходного значения канала №8	В режиме дискретного входа: 0...65535 = значение игнорируется;
Регистр выходного значения канала №9	В режиме дискретного выхода: ≥ 1 = установить на соответствующем контакте контроллера электрический потенциал заданный

Регистр выходного значения канала №10	0	конфигурацией данного канала для состояния «ВКЛ»; = установить на соответствующем контакте контроллера электрический потенциал заданный конфигурацией данного канала для состояния «ВЫКЛ».
Регистр включения активного режима / счетчика входа канала №8	0...65535	В режиме дискретного выхода: = значение игнорируется;
Регистр включения активного режима / счетчика входа канала №9	$\geq 1$	В режиме дискретного входа: = перевести соответствующий канал контроллера в активный режим работы / включить режим счетчика - реагировать на изменения потенциалов на соответствующем контакте контроллера / вести подсчет количества изменений потенциала из состояния «ВЫКЛ» в состояние «ВКЛ»;
Регистр включения активного режима / счетчика входа канала №10	0	= установить соответствующему каналу контроллера обычный (пассивный) режим работы в качестве дискретного входа.

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.

Регистр	Возвращаемые значения	
Регистр активности ввода-вывода	1	= 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;
	0	= за прошедший цикл выполнения функциональной диаграммы изменений на каналах контроллера или изменений значений регистров драйвера не обнаружено.
Регистр входного значения канала №8	1	В режиме дискретного входа или выхода: =



Регистр входного значения канала №9 Регистр входного значения канала №10	0	на соответствующем контакте контроллера установлен электрический потенциал соответствующий полярности данного канала в режиме «ВКЛ»; = на соответствующем контакте контроллера установлен электрический потенциал не соответствующий полярности данного канала в режиме «ВКЛ».
Регистр значения счетчика канала №8 Регистр значения счетчика канала №9 Регистр значения счетчика канала №10	0...65535 0	В режиме входа-счетчика: = число импульсов зарегистрированных на соответствующем контакте контроллера с момента активации режима счетчика; В остальных режимах: = значение всегда равно 0.

### 3.3 Состояние канала в момент запуска контроллера

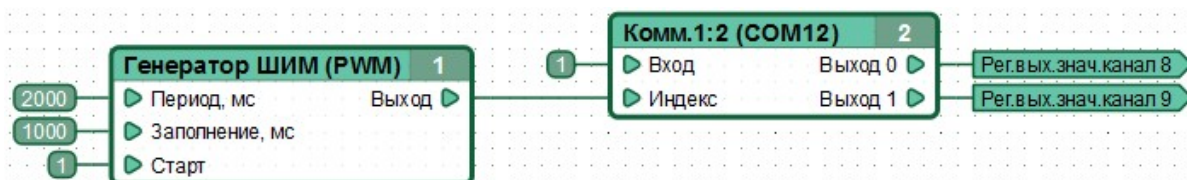
При включении контроллера, все силовые выходные элементы каналов находятся в выключенном состоянии. На контактах контроллера соответствующих каналам 8 и 9 устанавливаются потенциал 12В/1мА на контакте соответствующем каналу 10 — очень слабый отрицательный потенциал («воздух»).

### 3.4 Работа с каналами из пользовательской диаграммы

При записи ненулевого значения в регистр выходного значения канала на соответствующем контакте контроллера установится электрический потенциал «GND» 120мА, а при записи значения «0», потенциал соответствующий состоянию канала в момент запуска контроллера (см.выше).

*Примечание: В любой момент времени, вне зависимости от состояния выхода канала возможно получить значение, соответствующее фактическому текущему электрическому потенциалу на контакте данного канала, что позволяет организовать обратную связь и, при необходимости, реализовать программную защиту от перегрузки канала.*

Пример функциональной диаграммы выполняющей ежесекундное поочередное переключение электрических потенциалов с «+12В»(1мА) на «GND»(120мА) на контактах контроллера, соответствующих каналам №8 и №9.



**ВНИМАНИЕ!** В отличие от CANNY 7, каналы CANNY 5.2 duo не имеют автоматической 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 либо организуйте программную защиту канала в диаграмме!

Каналы возвращают значение «1» в регистре своего входного значения, если на соответствующем контакте контроллера установился электрический потенциал «GND»; и возвращает значение «0» в регистре входного значения, когда на соответствующем контакте контроллера установился электрический потенциал «+12В».

Изменение потенциала на контактах каналов №№8 и 9 приведет к немедленному автоматическому выходу контроллера из режима пониженного энергопотребления при нахождении контроллера в нём.

Каналы контроллера в режиме дискретного входа работают с максимальной чувствительностью, т.е. «защита от дребезга» не предусмотрена и должна быть реализована, при необходимости, в функциональной диаграмме.

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:

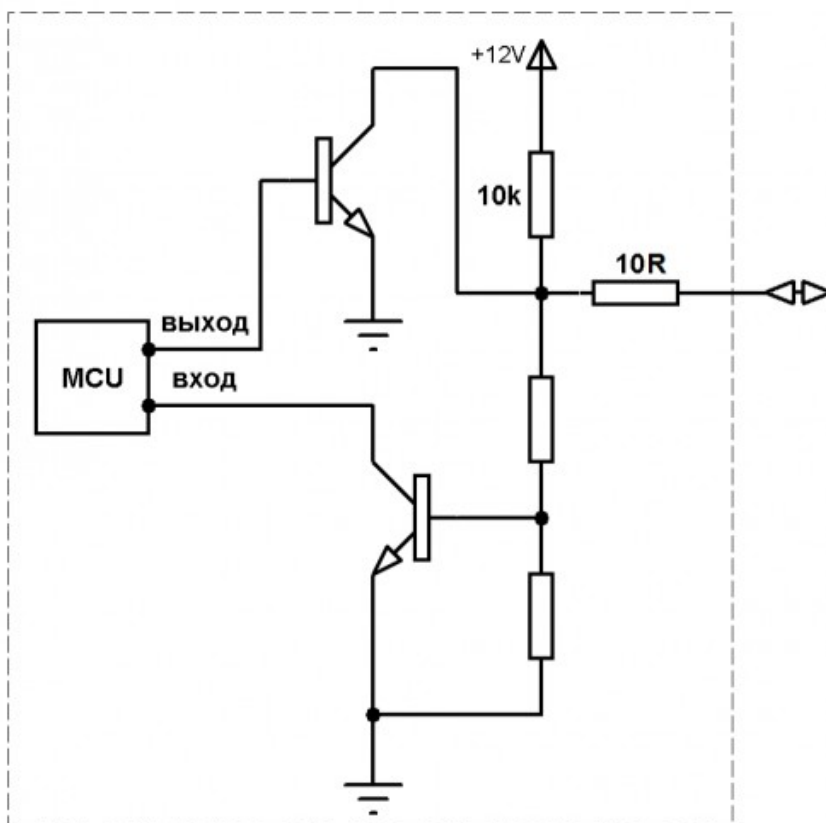


### 3.5 Эквивалентные принципиальные электрические схемы

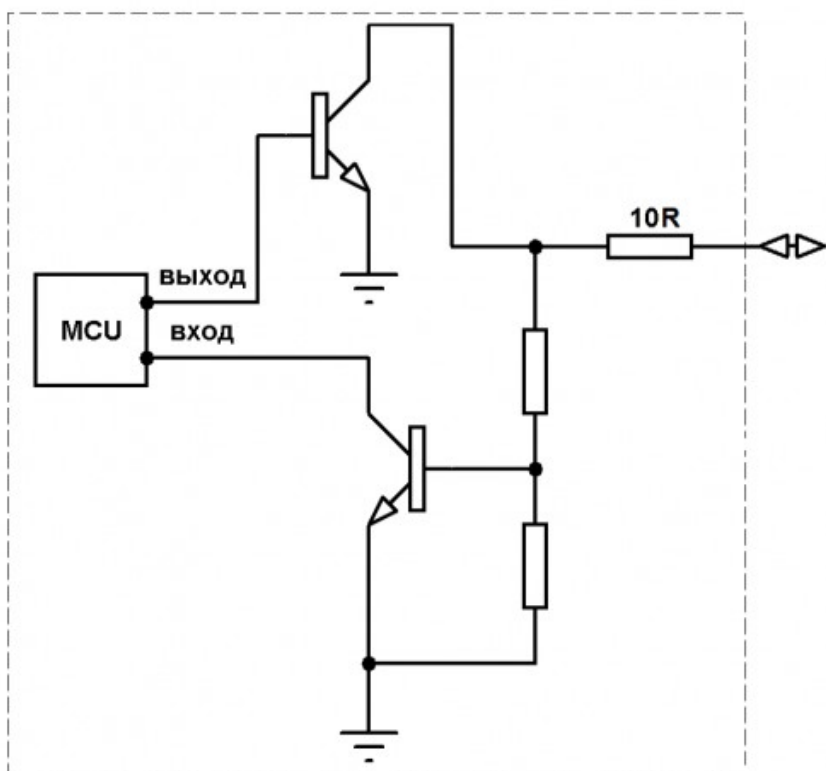
Эквивалентные электрические принципиальные схемы для каждого возможного состояния канала ввода-вывода контроллера приведены в таблице:

Состояние канала	Эквивалентная электрическая принципиальная схема
------------------	--

Канал №8  
Канал №9



Канал №10



# 4 CANNY 5.2 duo, Драйвер CAN

## 4.1 Общее описание

Контроллеры CANNY 5.2 duo имеют два независимых интерфейса CAN. Два специальных контакта разъема X1, используемые драйвером интерфейса CAN0 (CAN0-H и CAN0-L), и контакты разъема X2, используемые драйвером интерфейса CAN1 (CAN1-H и CAN1-L), предназначены для подключения к цифровой информационной **шине CAN**.

Особенностью драйвера CAN контроллера CANNY 5.2 duo является то, что количество фильтров принимаемых сообщений интерфейса CAN0, как и у CANNY 5.2, равно 8 (против 16 у CANNY 7), а у интерфейса CAN1 их всего 4.

Интерфейсы CAN0 и CAN1 имеют индивидуальные настройки и могут работать на разных скоростях обмена данными, что позволяет подключать контроллер CANNY 5.2 duo одновременно к двум разным шинам CAN.

## 4.2 Регистры драйвера

Ниже приведено описание допустимых значений регистров управления работой драйверов CAN, где x - номер интерфейса CAN, 0 или 1.

Регистры конфигурации драйвера CAN позволяют установить параметры работы соответствующего интерфейса CAN в качестве узла сети CAN и, при необходимости, ограничить объем принимаемых узлом сообщений при помощи фильтров:

Регистр	Ожидаемые значения
Регистр установки конфигурации CANx	1...N = активизация драйвера и установка скорости приема/передачи CAN-сообщений соответствующего интерфейса. Задается специальной константой из справочника констант; 0 = драйвер отключен.
Регистр установки фильтра приема CANx IDL №0 ... Регистр установки фильтра приема CANx IDL №7	0...0xFFFF = 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 приема CANx IDH №0 ... Регистр установки фильтра приема CANx IDH №7	0...0x1FFF = установить значение фильтра для старшей части идентификатора CAN-сообщения (биты 16...28 идентификатора расширенного формата);	
Регистр установки режима пассивного приема сообщений CANx	≥ 1	= включен режим пассивного приема (listen only) сообщений CAN соответствующего интерфейса;
	0	= включен режим нормального приема-передачи (normal) сообщений CAN соответствующего интерфейса;
Регистр режима фильтрации приема данных CANx	≥ 1	= режим фильтрации принимаемых сообщений CAN соответствующего интерфейса включен;
	0	= режим фильтрации принимаемых сообщений CAN соответствующего интерфейса отключен.
Регистр запрета автоматической повторной отправки сообщения CAN0 при ошибке	≥ 1	= режим запрета автоматической повторной отправки сообщения через интерфейс CAN0 при ошибке включен, при передаче сообщений, драйвером будет предприниматься только одна попытка отправить каждое сообщение CAN; Для интерфейса CAN1 этот параметр всегда установлен.
	0	= режим запрета автоматической повторной отправки сообщения через интерфейс CAN0 при ошибке отключен, отправка каждого сообщения будет автоматически повторяться до тех пор, пока интерфейсом не будет получено подтверждение приема сообщения 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.*

Именованные константы, определяющие конфигурацию интерфейсов 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.

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

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

Регистр	Возвращаемые значения	
Регистр наличия принятых данных CANx	1	= в буфере приема соответствующего интерфейса драйвера CAN находится полученное сообщение, данное значение появляется в регистре на один цикл выполнения диаграммы сообщая об

	<p>актуальности данных, находящихся в буфере приема;</p> <p>0 = в буфере приема соответствующего интерфейса драйвера CAN нет актуальных данных.</p>
Регистр принятого сообщения CANx IDL	0...0xFFFF = значение младшей части идентификатора полученного CAN-сообщения соответствующего интерфейса.
Регистр принятого сообщения CANx IDH	0...0x1FFF = значение старшей части идентификатора полученного CAN-сообщения соответствующего интерфейса.
Регистр принятого сообщения CANx ERL	0...0xC000 = значение, равное количеству байт данных в принятом сообщении CAN соответствующего интерфейса, признаки EXT и RTR (см. примечание).
Регистр принятого сообщения CANx D1:D0 ... Регистр принятого сообщения CANx D7:D6	0...0xFFFF = значения соответствующих байт данных принятого сообщения CAN указанного интерфейса, 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.*

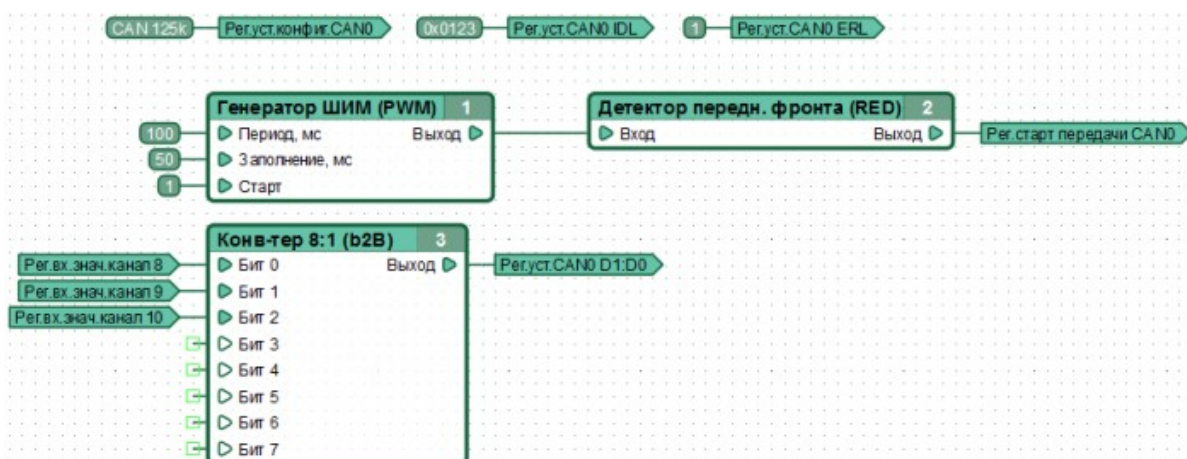
Регистры передачи сообщений интерфейсов CAN используются для размещения в буфере передачи драйвера данных, подлежащих отправке.

Регистр	Ожидаемые значения
Регистр начала передачи CANx	<p><math>\geq 1</math> = загрузить данные из регистров передачи в буфер передачи соответствующего интерфейса драйвера CAN;</p> <p>0 = не загружать данные в буфер передачи соответствующего интерфейса драйвера CAN.</p>
Регистр сообщения передачи CANx IDL	0...0xFFFF = значение младшей части идентификатора передаваемого CAN-сообщения соответствующего интерфейса.
	0...0x1FFF =

Регистр сообщения передачи CANx IDH	значение старшей части идентификатора передаваемого CAN-сообщения соответствующего интерфейса.
Регистр сообщения передачи CANx ERL	0...0xC000 = значение, равное количеству байт данных в передаваемом сообщении соответствующего интерфейса драйвера CAN, признаки EXT и RTR (см. примечание).
Регистр сообщения передачи CANx D1:D0 ... Регистр сообщения передачи CANx D7:D6	0...0xFFFF = значения соответствующих байт данных передаваемого сообщения CAN соответствующего интерфейса драйвера, по два байта на регистр.

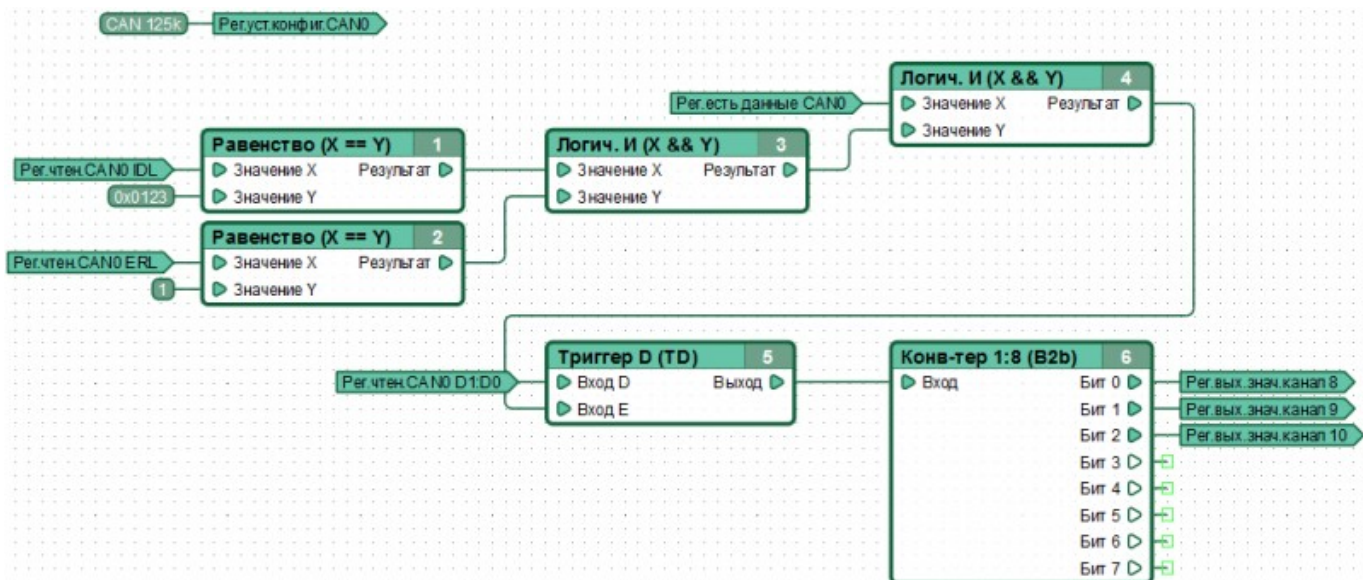
*Примечание: Регистр сообщения передачи CAN ERL, помимо числа байт в передаваемом сообщении 0...8 в младших битах, содержит в своих старших битах информацию о специальных признаках сообщения: бит 15 - признак EXT и бит 14 признак RTR. Где EXT = 1 при передаче сообщения в расширенном формате, EXT = 0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

Пример функциональной диаграммы отправки данных в шину CAN через интерфейс CAN0. 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.



Пример функциональной диаграммы приема данных из шины CAN через интерфейс CAN0. 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.





# 5 CANNY 5.2 duo, Драйвер шлюза CAN

## 5.1 Общее описание

Используя драйвер шлюза CAN возможно организовать высокоскоростную асинхронную ретрансляцию сообщений между аппаратными CAN-интерфейсами контроллера с возможностью автоматической модификации ретранслируемых сообщений.

Драйвер шлюза CAN позволяет организовать индивидуальную обработку полученных сообщений CAN путем настройки восьми правил отбора.

Каждое правило позволяет отбирать сообщения CAN по следующим, задаваемым пользователем признакам:

- номер интерфейса CAN по которому сообщение поступило в контроллер;
- идентификатор поступившего сообщения (Значение отбора).

При поступлении сообщения удовлетворяющего заданным признакам, драйвер позволяет выполнить одно из следующих действий:

- ретранслировать сообщение на другой интерфейс CAN контроллера без изменений;
- модифицировать полученное сообщение, а затем ретранслировать сообщение на другой интерфейс CAN контроллера;
- отбросить сообщение.

Модификация сообщения производится путем замены по маске любых бит исходного сообщения включая значения его идентификатора, длины и байт данных.

Для всех сообщений, поступивших с определенного интерфейса CAN, но не удовлетворяющих признакам ни одного из включенных правил отбора шлюза возможно задать одно из следующих действий:

- ретранслировать сообщение на другой интерфейс CAN контроллера без изменений;
- отбросить сообщение.

Для работы драйвера шлюза CAN необходимо задать конфигурацию драйверов обоих CAN-интерфейсов контроллера, при этом заданные для интерфейсов конфигурации CAN могут отличаться.

При включенном драйвере шлюза CAN, средствами пользовательской диаграммы возможно, используя регистры Драйвера CAN0 и Драйвера CAN1:

- отправлять сообщения через любой из CAN-интерфейсов контроллера в обычном режиме;
- получать сообщения CAN удовлетворяющие правилам отбора установленным для соответствующего интерфейса CAN драйвером шлюза CAN.

В процессе работы пользовательской диаграммы возможно многократно изменять любые параметры конфигурации драйвера шлюза CAN средствами пользовательской диаграммы, организуя таким образом, с некоторыми временными ограничениями, динамический режим работы шлюза.

## 5.2 Регистры драйвера

Ниже приведено описание допустимых значений регистров управления работой драйвера шлюза CAN, позволяющих установить параметры работы контроллера для высокоскоростной асинхронной ретрансляции сообщений между аппаратными CAN-интерфейсами.

Общие регистры драйвера шлюза CAN.

Регистр	Ожидаемые значения	
Регистр включения драйвера шлюза CAN	$\geq 1$	= драйвера шлюза включен;
	= 0	= драйвер шлюза отключен.
Регистр обновления конфигурации шлюза CAN	$\geq 1$	= применить новые настройки драйвера шлюза CAN (см.примечание);
	= 0	= не обновлять настройки драйвера шлюза CAN.
Регистр включения фильтрующего режима шлюза CAN для интерфейса CANx	$\geq 1$	= включить фильтрующий режим шлюза CAN для соответствующего интерфейса: ретранслировать на второй CAN-интерфейс контроллера и передавать в функциональную диаграмму только

		те CAN-сообщения, которые попали в отборы шлюза, остальные сообщения отбрасывают;
	= 0	= фильтрующий режим шлюза CAN для соответствующего интерфейса отключен: автоматически ретранслировать не попавшие в отборы шлюза сообщения на второй CAN-интерфейс контроллера в неизменном виде.
Регистр включения режима логирования шлюза CAN для интерфейса CANx	≥ 1	= включить режим логирования шлюза CAN для соответствующего интерфейса: передавать в функциональную диаграмму (в драйвер CANx) все принимаемые по данному интерфейсу CAN-сообщения, при этом работа драйвера шлюза в части отборов, замены и ретрансляции сообщений CAN осуществляется согласно соответствующим настройкам шлюза указанного CAN-интерфейса;
	= 0	= режим логирования шлюза CAN для соответствующего интерфейса отключен: в диаграмму (в драйвер CANx) передаются только CAN-сообщения, которые попали в отборы шлюза, остальные сообщения отбрасываются.

Регистры отбора драйвера шлюза CAN.

Общие регистры отбора драйвера шлюза CAN.

Регистр	Ожидаемые значения	
Регистр включения отбора X шлюза CAN	≥ 1	= включить соответствующий отбор шлюза CAN;
	= 0	= соответствующий отбор шлюза CAN отключен.
Регистр установки номера CAN-интерфейса отбора X шлюза CAN	= 0	= применять соответствующий отбор к сообщениям, поступающим в контроллер через интерфейс CAN0;
	= 1	= применять соответствующий отбор к сообщениям, поступающим в контроллер через интерфейс CAN1.
Регистр установки запрета ретрансляции отбора X шлюза CAN	≥ 1	= отбрасывать все сообщения удовлетворяющие значению соответствующего отбора;
	= 0	=

	разрешить обработку и автоматическую ретрансляцию на другой CAN интерфейс всех сообщений удовлетворяющих значению соответствующего отбора.
--	--

Регистры значений отборов драйвера шлюза CAN.

**Значение отбора** - параметр, при удовлетворении которому, поступившие CAN-сообщения отбираются из общего потока данных для специальной обработки.

Регистр	Ожидаемые значения
Регистр установки значения отбора X шлюза, CAN IDL	0...0xFFFF = значение младшей части идентификатора (биты 0...10 идентификатора стандартного формата или биты 0...15 идентификатора расширенного формата) CAN-сообщения удовлетворяющего соответствующему отбору.
Регистр установки значения отбора X шлюза, CAN IDH	0...0x1FFF = значение старшей части идентификатора (биты 16...28 идентификатора расширенного формата) CAN-сообщения удовлетворяющего соответствующему отбору.

Регистры масок отборов драйвера шлюза CAN.

**Маска отбора** - группа значений, биты которых определяют подлежащие модификации при ретрансляции шлюзом части сообщения попавшего в соответствующий отбор. При значении бита маски равном "0", соответствующий бит принятого сообщения будет ретранслирован на другой интерфейс без изменений. При значении бита маски равном "1", соответствующий бит принятого сообщения будет заменен при ретрансляции соответствующим битом из значения *Замены отбора*

Регистр	Ожидаемые значения
Регистр установки маски отбора X шлюза, CAN IDL	0...0xFFFF = установить значение маски младшей части идентификатора CAN-сообщения (биты 0...10 идентификатора стандартного формата или биты 0...15 идентификатора расширенного формата) соответствующего отбора.
Регистр установки маски отбора X шлюза, CAN IDH	0...0x1FFF = установить значение маски старшей части идентификатора CAN-сообщения (биты 16...28 идентификатора расширенного формата) соответствующего отбора.

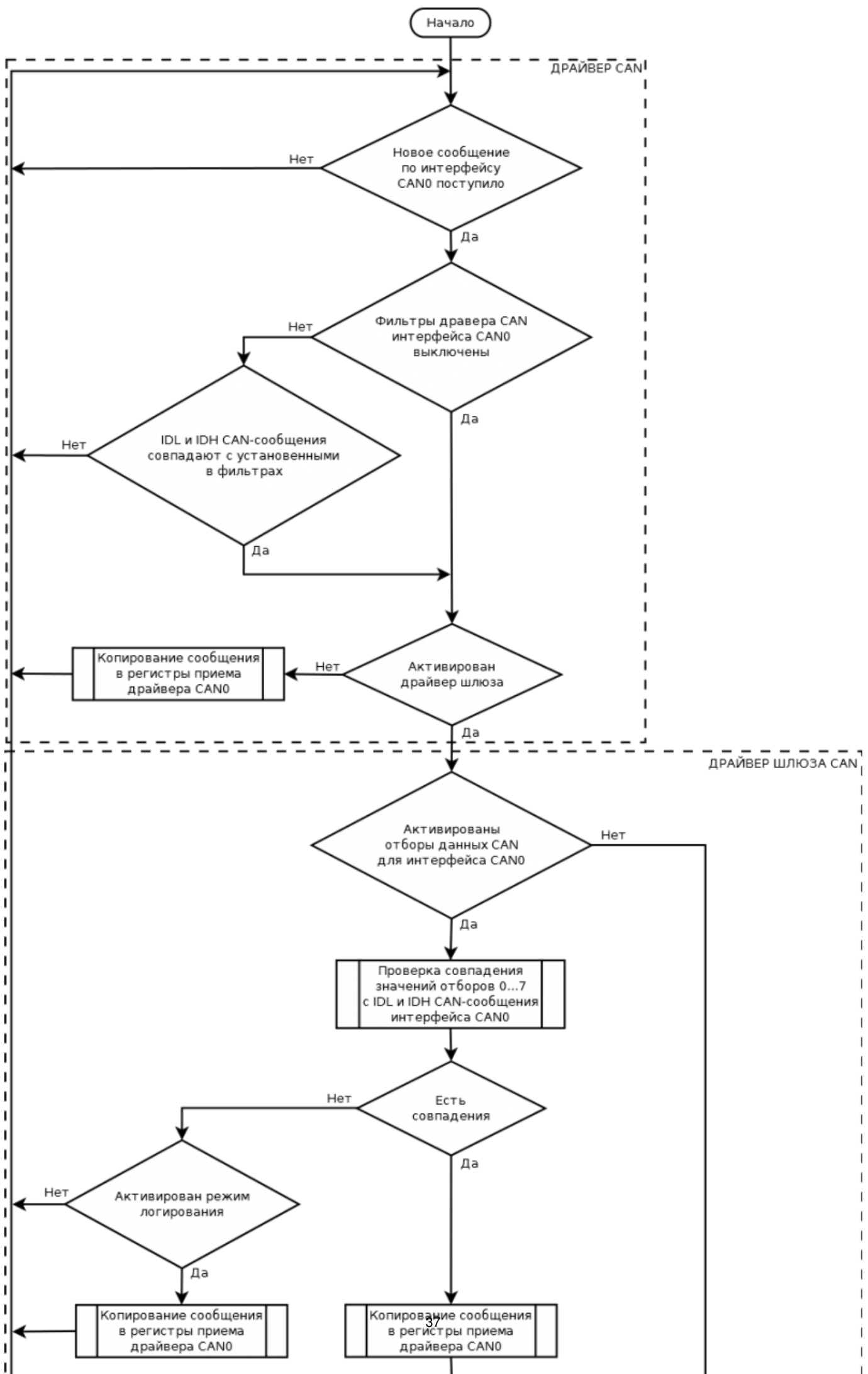
Регистр установки маски отбора X шлюза, CAN ERL	0...0xFFFF = установить значение маски регистра ERL CAN-сообщения соответствующего отбора.
Регистр установки маски отбора X шлюза, CAN D1:D0 ... Регистр установки маски отбора X шлюза, CAN D7:D6	0...0xFFFF = установить значение маски соответствующих байт данных CAN-сообщения соответствующего отбора.

Регистры замены отборов драйвера шлюза CAN.

**Замена отбора** - группа значений, биты которых определяют результат модификации попавшего в соответствующий отбор ретранслируемого сообщения. При значении бита замены равном "0", соответствующий бит ретранслируемого сообщения будет заменен нулем, при значении бита замены равном "1", соответствующий бит ретранслируемого сообщения будет заменен единицей. Замена будет применена лишь к тем битам, значение маски отбора для которых равно "1".

Регистр	Ожидаемые значения
Регистр установки замены отбора X шлюза, CAN IDL	0...0xFFFF = установить значение замены младшей части идентификатора CAN-сообщения (биты 0...10 идентификатора стандартного формата или биты 0...15 идентификатора расширенного формата) соответствующего отбора.
Регистр установки замены отбора X шлюза, CAN IDH	0...0x1FFF = установить значение замены старшей части идентификатора CAN-сообщения (биты 16...28 идентификатора расширенного формата) соответствующего отбора.
Регистр установки замены отбора X шлюза, CAN ERL	0...0xFFFF = установить значение замены регистра ERL CAN-сообщения используемого отбора.
Регистр установки замены отбора X шлюза, CAN D1:D0 ... Регистр установки замены отбора X шлюза, CAN D7:D6	0...0xFFFF = установить значения замены соответствующих байт данных CAN-сообщения соответствующего отбора.

### **5.3 Блок-схема работы контроллера в режиме CAN-шлюза**







# 6 CANNY 5.2 duo, Драйвер UART

## 6.1 Общее описание

Драйвер **UART**, через USB Virtual COM-port, позволяет контроллеру CANNY 5.2 duo обмениваться данными с внешними устройствами, например ПК, в процессе выполнения пользовательской диаграммы.

Объем приемного и передающего буферов данных драйвера UART CANNY 5.2 duo составляет 24 байта.

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

*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").*

*Примечание: Для возможности обмена данными между контроллером и внешним устройством, например ПК, через USB Virtual COM-port необходимо, при задании настроек драйвера использовать **только конфигурацию RS-232 (инверсная полярность)**.*

## 6.2 Регистры драйвера

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

Регистры конфигурации драйвера UART.

Регистр	Ожидаемые значения
Регистр конфигурации UART1	1...N = установить конфигурацию драйвера UART контроллера, определяющую текущий режим и параметры его работы (задается специальной константой из справочника констант); 0 = отключить драйвер UART.
Регистр установки таймаута приема сообщения UART1, мс	1...65535 = прекращение приема данных, если в течении указанного времени на линии не было зафиксировано ни одного изменения потенциала и линия находится в пассивном состоянии; 0 = использовать значение по умолчанию,

	задаваемое в конфигурации канала (13).	
Регистр включения режима открытого коллектора канала UART1 TX	≥ 1	= отключить верхнее плечо канала передачи интерфейса UART, передача будет осуществляться в режиме «воздух / GND»;
	0	= включить верхнее и нижнее плечо канала передачи интерфейса UART передача будет осуществляться в режиме «+5B / GND».

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

Параметр	Перечень допустимых значений
Скорость передачи данных, бод	300, 1200, 2400, 9600, 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.

*Примечание: Для возможности обмена данными между контроллером и внешним устройством, например ПК, через USB Virtual COM-port необходимо, при задании настроек драйвера использовать **только конфигурацию RS-232 (инверсная полярность)**.*

Регистры диагностики драйвера UART.

Регистр	Возвращаемые значения	
Регистр переполнения буфера UART1	1	= буфер UART переполнен;
	0	= переполнения не зафиксировано.
Регистр отсутствия активности драйвера UART1	1	= активность соответствующего канала драйвера UART отсутствует, линия находится в пассивном режиме;
	0	= зафиксирована активность на линии соответствующего канала драйвера UART.
Регистр ошибки приема	1	= во время приема данных UART произошла

UART1	0	ошибка; = драйвер работает в нормальном режиме.
Регистр готовности буфера передачи данных UART1	1	= буфер передачи данных драйвера UART свободен;
	0	= буфер передачи данных драйвера UART занят, передача данных невозможна.

Регистры приема драйвера UART.

Регистр	Возвращаемые значения	
Регистр наличия принятых данных UART1	1	= сообщение получено и помещено в буфер приема соответствующего канала драйвера UART;
	0	= в буфере приема соответствующего канала драйвера UART отсутствуют актуальные данные.
Регистр длины принятого сообщения UART1	0...24	= значение, равное количеству байт данных, в принятом по соответствующему каналу драйвера UART пакете данных.
Регистр принятого сообщения UART1 D1:D0 ... Регистр принятого сообщения UART1 D23:D22	0...0xFFFF = значения соответствующих байт данных приемных буферов UART каждого канала, по два байта на регистр.	

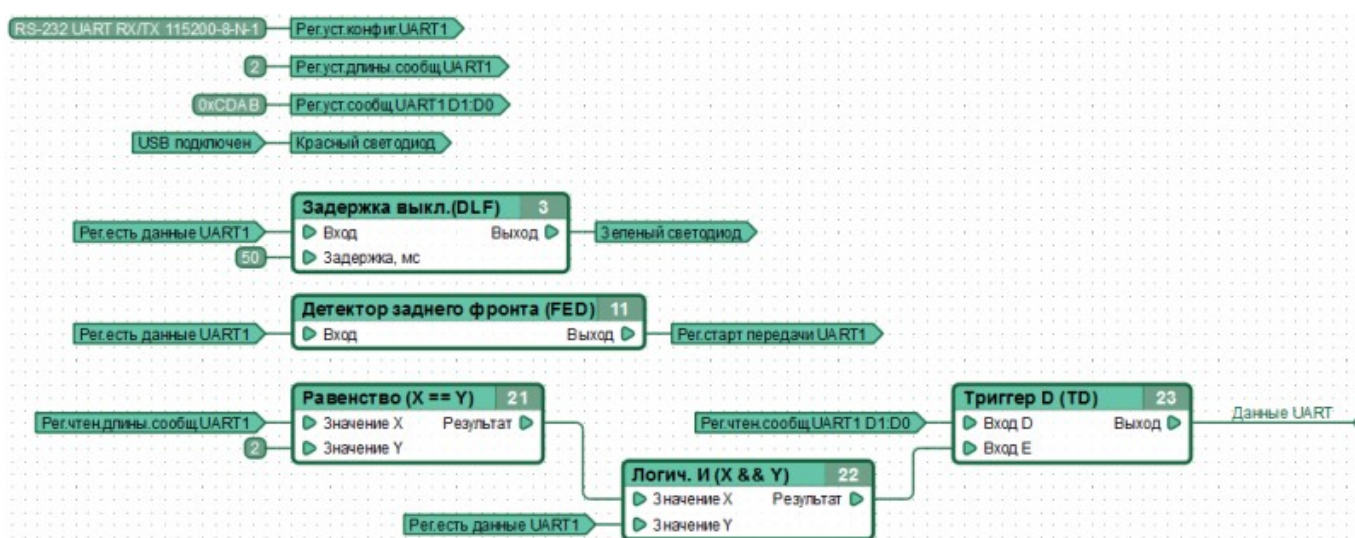
Регистры передачи драйвера UART.

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...24	= 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 D23:D22	0...0xFFFF = значения соответствующих байт данных для передачи по соответствующему каналу драйвера UART, по два байта на регистр.	

## 6.3 Работа контроллера в режиме UART

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

Пример функциональной диаграммы для обмена данными по UART через Virtual COM-port. 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. Подключение к ПК, через разъем USB, подтверждается включением красного светодиода контроллера. Получение данных по UART контроллер подтверждает кратковременным включением зеленого контрольного светодиода.



# 7 CANNY 5.2 duo, Параметры пользовательской конфигурации

## 7.1 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.

## 7.2 Регистры параметров пользовательской конфигурации

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

Регистр	Ожидаемые значения
Регистр параметра пользовательской конфигурации №0 ... Регистр параметра пользовательской конфигурации №15	0...65535 = значение соответствующего пользовательского параметра.

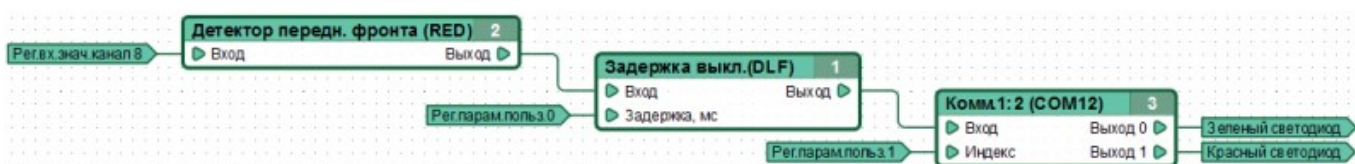
Значения в данных регистрах установятся при старте контроллера, после записи в него ПО посредством исполняемого файла автономной загрузки, и будут оставаться неизменными (константными) на протяжении всего времени работы функциональной диаграммы, не изменяясь даже при сбросе контроллера. Изменить значения данных регистров можно лишь стерев или перезаписав память контроллера новым ПО.

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

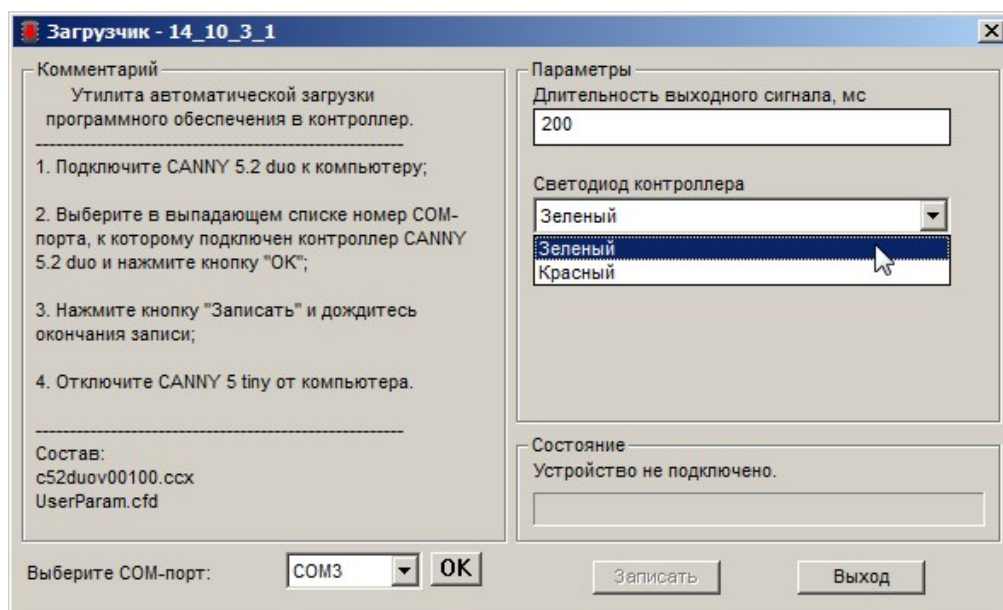
*Примечание: При записи контроллера из среды CannyLab значения всех регистров параметров пользовательской конфигурации устанавливается равным нулю.*

## 7.3 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 пользовательских параметра: Имя «Длительность выходного сигнала,мс», Тип «Число» и Имя «Светодиод контроллера», Тип «Список». В список значений параметра «Выходной канал контроллера» добавьте две строки: Название «Зеленый», Значение «0» и «Красный», Значение «1».



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



Выполняя диаграмму контроллер, в момент получения на входе канала №8 отрицательного потенциала, включает светодиод контроллера заданного пользователем в соответствующем параметре цвета и удерживает его заданное пользователем время.

# 8 CANNY 5.2 duo, Энергонезависимая память (ЭНП)

## 8.1 Общее описание

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

Пользователю доступны 64 шестнадцатибитные 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.*

## 8.2 Регистры энергонезависимой памяти

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

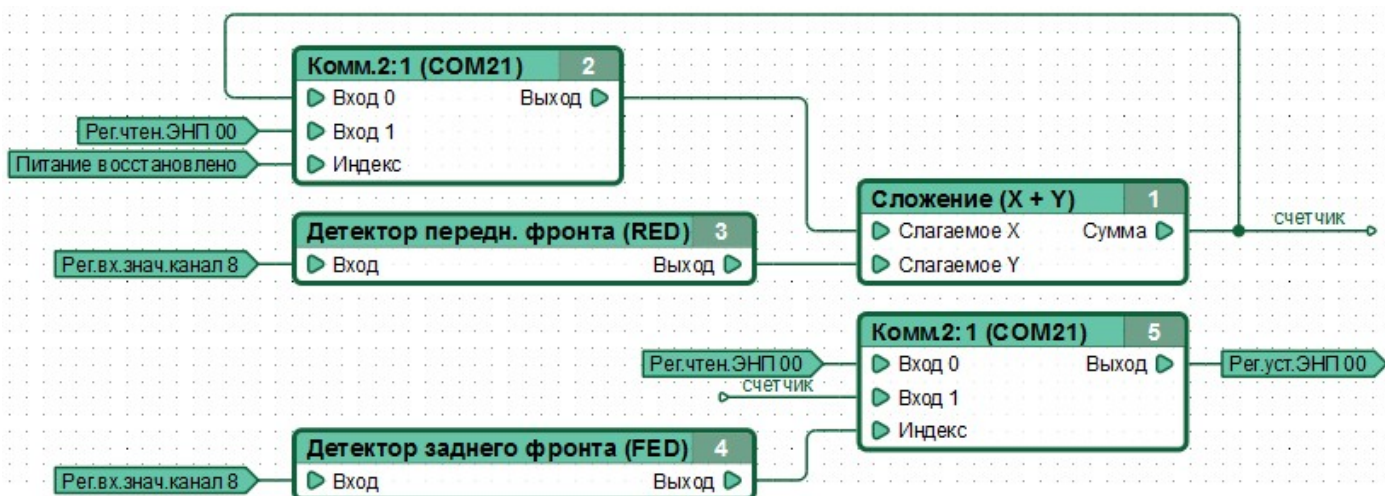
Регистр	Ожидаемые значения
Регистр установки энергонезависимой памяти №00 ... Регистр установки энергонезависимой памяти №63	0...65535 = сохраняемое значение.

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

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



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 отпуске 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 duo 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).*