

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/>**

1 CANNY 5 nano	1
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
1.6 Принципиальная схема CANNY 5 nano	16
2 CANNY 5 nano, Системные ресурсы и режимы работы	17
2.1 Общее описание	17
2.2 Сброс контроллера	17
2.3 Режим пониженного энергопотребления	18
2.4 Изменение разрешения системного таймера	20
2.5 Фактическое время выполнения функциональной диаграммы	21
2.6 Идентификатор устройства	22
2.7 Контроль активности интерфейсов контроллера	23
2.8 Идентификатор вендора устройства	23
3 CANNY 5 nano, Драйвер каналов ввода-вывода	25
3.1 Общее описание	25
3.2 Особенности работы канала №13	25
3.3 Регистры драйвера	26
3.4 Состояние канала в момент запуска контроллера	29
3.5 Режим дискретного выхода	29
3.6 Режим дискретного входа	30
3.7 Режим счетчика	31
4 CANNY 5 nano, Драйвер широтно-импульсного модулятора (ШИМ)	33
4.1 Общее описание	33
4.2 Регистры драйвера	34
4.3 Особенности работы	35
5 CANNY 5 nano, Драйвер UART	37
5.1 Общее описание	37
5.2 Регистры драйвера	37
5.3 Работа контроллера в режиме UART	40

6 CANNY 5 nano, Драйвер CAN.....	41
6.1 Общее описание.....	41
6.2 Регистры драйвера.....	41
7 CANNY 5 nano, Драйвер I2C.....	46
7.1 Общее описание.....	46
7.2 Регистры драйвера I2C.....	46
7.3 Особенности работы драйвера I2C.....	48
8 CANNY 5 nano, Драйвер Dallas 1-Wire.....	50
8.1 Общее описание.....	50
8.2 Регистры драйвера 1-Wire.....	50
9 CANNY 5 nano, Параметры пользовательской конфигурации.....	55
9.1 Общее описание.....	55
9.2 Регистры параметров пользовательской конфигурации.....	55
9.3 Пример использования параметров пользовательской конфигурации.....	56
10 CANNY 5 nano, Энергонезависимая память (ЭНП).....	58
10.1 Общее описание.....	58
10.2 Регистры энергонезависимой памяти.....	58
11 CANNY 5 nano, Драйвер пульта ИК ДУ.....	60
11.1 Общее описание.....	60
11.2 Регистры драйвера пульта ИК ДУ.....	60
12 CANNY 5 nano, Аналого-цифровой преобразователь (АЦП).....	65
12.1 Общее описание.....	65
12.2 Регистры драйвера.....	65
13 CANNY 5 nano, Драйвер датчика DHT11-DHT22-AM2302.....	67
13.1 Общее описание.....	67
13.2 Регистры драйвера датчика DHT11-DHT22-AM2302.....	67

1 CANNY 5 nano

CANNY 5 nano — компактный программируемый логический контроллер ориентированный на автомобильное, бытовое и промышленное применение, предназначенный для макетирования и использования в качестве встраиваемого контроллера.

1.1 Общие сведения

Программируемый логический контроллер CANNY5 nano является встраиваемым ПЛК, обладающим широким функционалом.

По своим функциональным и конструктивным параметрам CANNY5 nano может быть отнесен к классу встраиваемых интеллектуальных реле или к классу электронных конструкторов.

К основным особенностям CANNY5 nano можно отнести:

- два варианта напряжения питания 6..15В или 5В;
- возможность обеспечения питанием внешних устройств: 5В до 800 мА
- номинальное напряжение каналов ввода-вывода 0 / 5В;
- 18 универсальных каналов ввода/вывода с максимальным током каждого из них 25мА и общим током не более 800мА;
- встроенный интерфейс CAN 2.0В совместимый с ISO-11898, SAE J2411 широко применяемым в автомобилях;
- поддержка популярных протоколов обмена данными: 2 UART, LIN*, I²C, Dallas, ИК ДУ;
- встроенный USB-UART преобразователь, используемый как для программирования и/или питания контроллера, так и для взаимодействия пользовательских диаграмм с ПК;
- восьмиканальный аналого-цифровой преобразователь (АЦП);
- 3 канала высокочастотного ШИМ, с разрешением 1мкс;
- встроенные средства управления собственным энергопотреблением контроллера в диапазоне от 0,5 до 20мА;
- энергонезависимая память программ и шестьдесят четыре 16-и битные ячейки энергонезависимой памяти данных доступные пользовательскому приложению, способные сохранить критически важные данные при сбоях питания контроллера;
- широкий диапазон рабочих температур от -40 до + 85 оС;
- компактное исполнение и возможность встраивания в качестве центрального управляющего процессора позволяет создавать уникальные пользовательские устройства.

(* - находится в разработке)

Для создания пользовательских программ CANNY5 nano, как и для программирования других контроллеров CANNY, используется графический язык программирования CFD. Разработка программ осуществляется в интегрированной среде 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 CANNY5 nano.

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

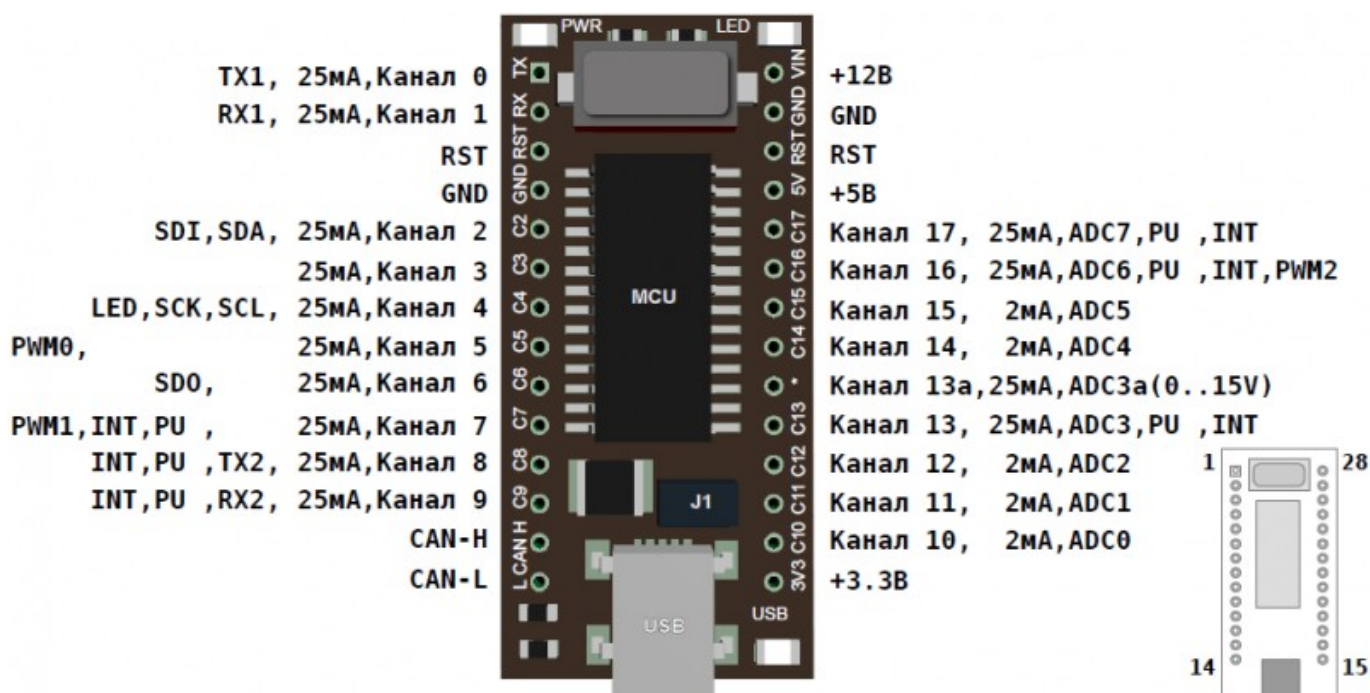
Режим работы восемнадцати внешних каналов контроллера может быть настроен индивидуально из пользовательской функциональной диаграммы.

1.2 Устройство и принцип работы

1.2.1 Внешний вид и расположение элементов

Основными конструктивными элементами CANNY5 nano являются: микроконтроллер (MCU), смонтированный на плате совместно с кварцевым резонатором, переключкой, разъемом miniUSB, на лицевой стороне платы контроллера; блока питания, контроллера USB, трансмиттера CAN - на оборотной. Плата контроллера укомплектована двумя разъемами PLS, по 14 контактов, расположенными по одному, вдоль каждой из ее длинных сторон. Данные разъемы могут быть использованы для установки контроллера на монтажную плату или в стандартную 28pin колодку. Переключение между рабочим режимом и режимом загрузки в контроллер программного обеспечения выполняется с помощью переключки J1 (джампера).

Для возможности отслеживания состояния контроллера и режимов его работы на лицевой стороне его платы установлены два контрольных светодиода с помощью которых отображается наличие питания на процессоре и активность соединения с ПК. Еще один светодиод указывает на наличие положительного потенциала на четвертом канале контроллера и может быть таким образом использован пользователем для индикации режимов работы контроллера из функциональной диаграммы.



1.2.2 Питание контроллера

В контроллерах CANNY5 nano допускается 3 варианта питания:

- от источника постоянного тока напряжением 6..15В: положительный выход источника питания подключается к контакту №28 платы контроллера, потенциал GND — к контакту №4 или №27;
- от источника постоянного тока напряжением 5В: потенциал +5В подключается к контакту №25 платы контроллера, потенциал GND — к контакту №4 или №27;
- напряжением 5В от USB-порта ПК, через разъем miniUSB на плате контроллера.

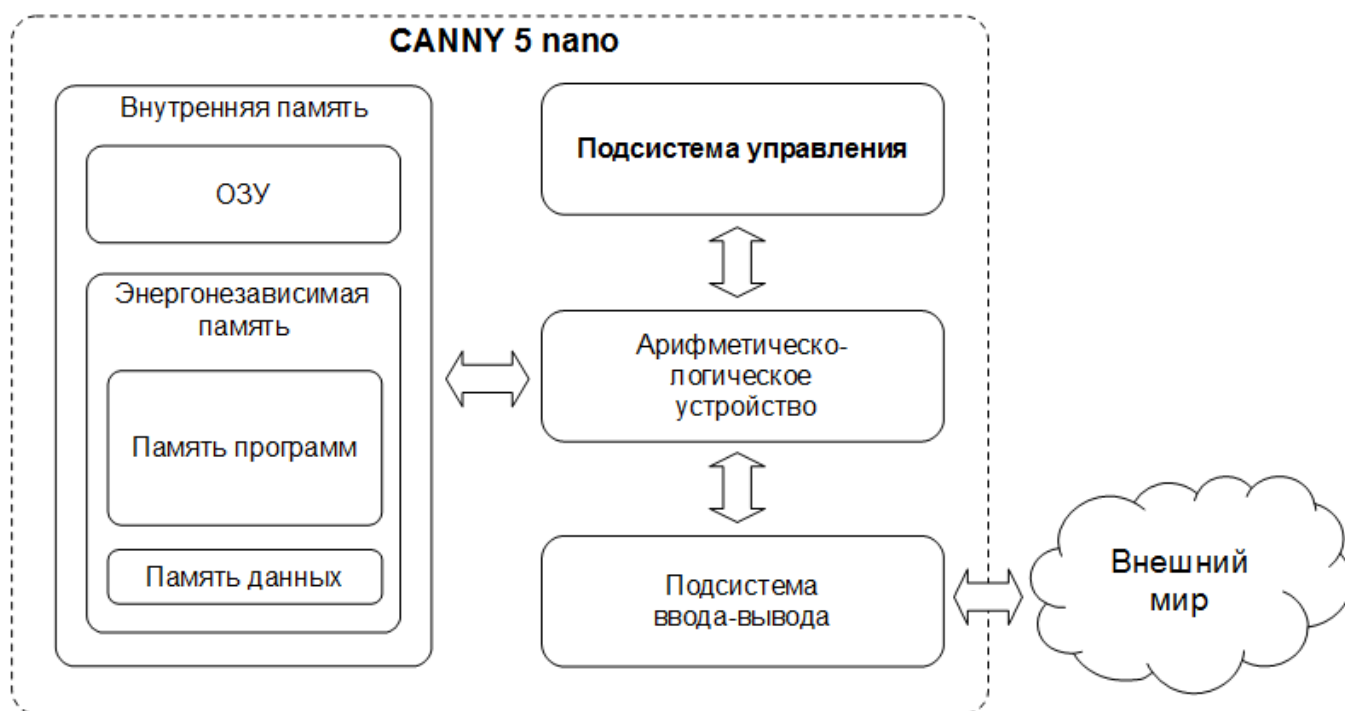
Примечание: При питании контроллера от источника постоянного тока через контакт №28 или через USB-разъем, контакте №25 может быть использован для питания внешних устройств напряжением 5В. При этом, совокупный ток потребления контроллера и внешних устройств не должен превышать 800 мА, а питании через USB 500мА. При питании от USB, внешние устройства могут так же получить питание напряжением 3.3В током до 50мА от контакта №15 контроллера.

1.2.3 Программная архитектура

CANNY5 nano является цифровым программируемым вычислительным управляющим устройством.

В целом, для CANNY5 nano справедливы общие сведения о программируемых логических контроллерах изложенные во введении к настоящему руководству.

Основными элементами CANNY5 nano являются: арифметическо-логическое устройство (АЛУ), внутренняя память, подсистема управления ходом исполнения команд и система ввода-вывода.



Арифметическо-логическое устройство — вычислительное ядро CANNY5 nano. АЛУ обеспечивает исполнение системного программного обеспечения и пользовательских функциональных диаграмм, помещенных во внутреннюю память контроллера.

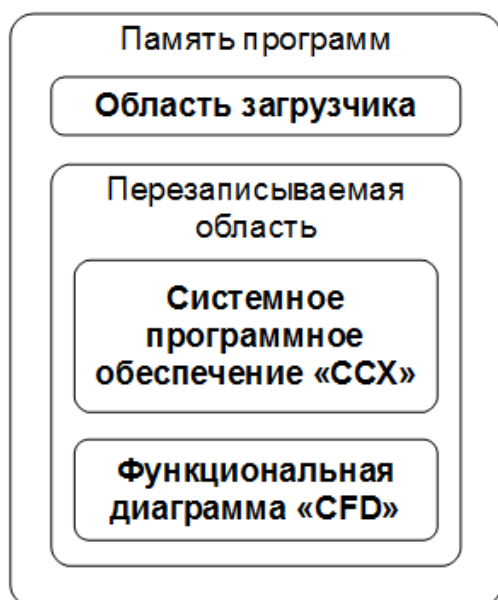
Внутренняя память контроллера разделяется на энергонезависимую память программ, энергонезависимую память данных и оперативную память данных.

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.

1.2.4 Структура программного обеспечения

Программное обеспечение CANNY5 nano состоит из: программного загрузчика, системного ПО (операционной системы) и пользовательской функциональной диаграммы.



Программный загрузчик обеспечивает работу контроллера в режиме загрузки ПО, осуществляя обмен данными между CANNY5 nano и персональным компьютером, выполняя 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 CANNY5 nano 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 память контроллера.

Пользовательская функциональная диаграмма создается и модифицируется пользователем с помощью интегрированной среды 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 память контроллера по командам среды разработки CannyLab. Вход в данный режим осуществляется при подключении контроллера к ПК с помощью кабеля-переходника USB-A в miniUSB, при отключенном внешнем питании и замкнутой перемычке/джампере J1. После успешного подключения контроллера к ПК пользователь может выполнить установку связи CANNY5 nano с интегрированной средой разработки CannyLab.

Примечание: Для корректной работы CANNY5 nano с операционной системой ПК может потребоваться установка специального драйвера, доступного для бесплатной загрузки с сайта производителя.

Выход из данного режима происходит при размыкании перемычки J1 на плате контроллера. Если в момент выхода из режима загрузки ПО, энергонезависимая память программ контроллера содержала корректно записанное системное программное обеспечение, то 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 по одному из доступных вариантов, при разомкнутой перемычке J1.

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.

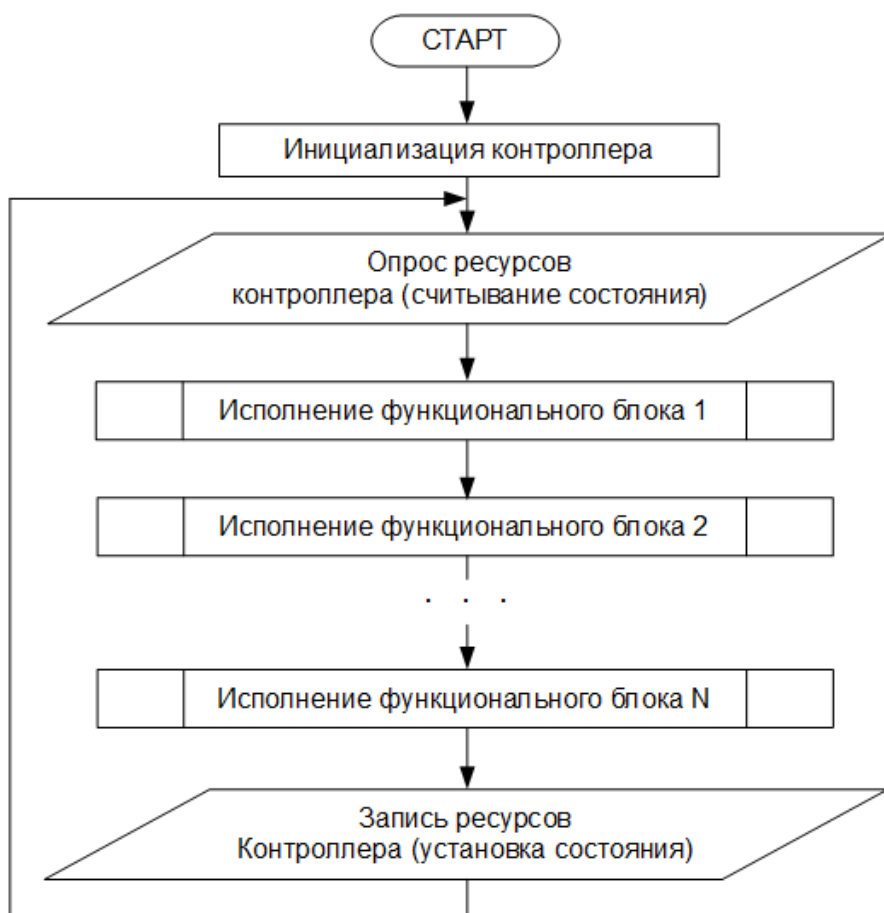
1.4 Среда исполнения функциональных диаграмм

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 CANNY5 nano.

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.

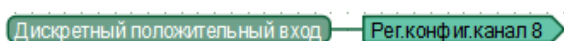
Пользователь имеет возможность указать регистр чтения в качестве источника входных данных практически любого функционального блока на диаграмме и, тем самым, извлечь и использовать при реализации собственных алгоритмов сведения, полученные контроллером из внешнего мира. Например, информацию об электрическом 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 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 «0» в регистр, расположенный по адресу 1228 задается режим работы канала №8 в качестве дискретного входа.

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».



Установив таким образом режим работы канала №8, мы можем по появлению значения «1» в регистре расположенном по адресу 1192 («Регистр входного значения канала №8»), узнать о приложении положительного электрического потенциала к контакту «C8» контроллера.



1.5 Ресурсы контроллера

1.5.1 Системные ресурсы и режимы работы

Основная статья: CANNY 5 nano, 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 CANNY Lab через контекстное меню элементов диаграммы типа «Регистр чтения» и «Регистр записи».

1.5.2 Драйвер каналов ввода-вывода

Основная статья: CANNY 5 nano, Драйвер каналов ввода-вывода

Пользователям CANNY5 nano доступны восемнадцать дискретных каналов ввода-вывода общего назначения. Каждый канал физически представлен соответствующим контактом платы контроллера. Записывая данные в соответствующие регистры каналов драйвера, функциональная диаграмма может управлять электрическим потенциалом на каждом из этих контактов. Считывая данные регистров каналов ввода-вывода, диаграмма может получать информацию о текущем значении потенциала каждого из них.

Физические характеристики каналов позволяют подключать к ним светодиоды или слаботочные внешние управляющие устройства, например мощные транзисторы или транзисторные сборки, которые позволяют передавать управляющие сигналы на исполнительные устройства — электромагнитные реле, небольшие электродвигатели. В качестве внешних источников дискретных сигналов способных управлять работой контроллера, возможно использовать механические, электромеханические и электронные кнопки и переключатели, генераторы импульсов, источники напряжения 0-5В и 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.

1.5.3 Драйвер широтно-импульсного модулятора (ШИМ)

Основная статья: CANNY 5 nano, Драйвер широтно-импульсного модулятора (ШИМ)

Три из восемнадцати каналов ввода-вывода (Канал №5, Канал №7 и Канал №16) CANNY5 nano поддерживают работу в режиме широтно-импульсного модулятора.

Использование драйвера ШИМ позволяет указанным каналам контроллера работать асинхронно функциональной диаграмме, что дает возможность добиться максимальной

стабильности временных параметров генерируемого сигнала.

Драйвер ШИМ CANNY5 nano имеет два основных режима работы, задаваемых отдельно для каждого из трех каналов: низкочастотный и высокочастотный. Низкочастотный режим, позволяет реализовывать импульсный режим работы соответствующего канала с периодом от 2 до 65536 миллисекунд с шагом 1 миллисекунда, при использовании стандартных настроек системного таймера, и периодом от 200 до 6553600 микросекунд с шагом 100 микросекунд, при использовании увеличенного разрешения системного таймера.

В режиме высокочастотного ШИМ, период задается в диапазоне от 2 до 256 микросекунд, с шагом 1 микросекунда вне зависимости от настроек масштаба времени системного таймера.

1.5.4 Драйвер UART

Основная статья: CANNY 5 nano, Драйвер UART

Контроллеры CANNY 5 nano имеют два независимых аппаратных интерфейса **UART** поддерживающих 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. Интерфейсы могут быть задействованы независимо друг от друга и иметь индивидуальные настройки скорости передачи данных и полярности сигналов. Интерфейс UART1 задействует каналы № 0 (TX) и № 1 (RX) контроллера, интерфейс UART2 задействует каналы № 8 (TX) и № 9 (RX). При включении интерфейса активируются оба его канала.

Объем приемного и передающего буферов данных драйвера UART CANNY 5 nano составляет 32 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").

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.

1.5.5 Драйвер CAN

Основная статья: **CANNY 5 nano, Драйвер CAN**

Два специальных контакта контроллера CANNY 5 nano, CAN-H и CAN-L, предназначены для подключения к цифровой информационной **шине CAN**.

Особенностью драйвера CAN контроллера CANNY 5 nano с драйвером CAN является то, что у него количество фильтров принимаемых сообщений CAN равно 8, против 16 у CANNY 7.

1.5.6 Драйвер I²C

Основная статья: **CANNY 5 nano, Драйвер I²C**

Два из восемнадцати каналов ввода-вывода CANNY 5 nano, а именно каналы №2 и №4, поддерживают работу в режиме приема/передачи данных, с использованием протокола **I²C**, и могут быть использованы для связи контроллеров с внешним оборудованием поддерживающим данный протокол связи.

Контроллер CANNY5 nano может выступать только в качестве ведущего узла сети, при этом он имеет возможность как передавать данные ведомым устройствам, так и отправлять запросы на получение данных от них. Обмен данными между устройствами происходит отдельными сеансами, с максимальной длиной сообщения I²C внутри одного сеанса равной 32 байтам, т. е. открытие одновременно несколько сеансов с разными устройствами не допускается. Скорость обмена фиксированная и составляет 100 кбит/с. Общее число ведомых устройств на линии может достигать нескольких десятков.

Драйвер I²C в своей работе использует ресурсы каналов контроллера, но имеет более высокий приоритет чем драйвер дискретного ввода-вывода. Таким образом, при активации драйвера I²C, для задействованных в его работе каналов, изменение значений в связанных с ними регистрах драйвера дискретного ввода-вывода будет проигнорировано контроллером.

Каналы №№2 и 4 активируются совместно, одновременно с установкой конфигурации драйвера I²C контроллера, т. е. его активацией. Канал №2 используется в качестве линии данных (SDA), канал №4 — как тактирующий канал (SCL).

Примечание: Активация драйвера I²C автоматически переводит каналы №2 и №4 в режим передачи/приема данных, делая невозможным их использование в качестве дискретных входов или выходов драйвера ввода-вывода, т. к. драйвер I²C имеет более высокий приоритет.

1.5.7 Драйвер Dallas 1-Wire

Основная статья: **CANNY 5 nano, Драйвер Dallas 1-Wire**

Контроллер CANNY 5 nano может быть использован в качестве ведущего (MASTER) узла в однопроводной сети передачи данных **Dallas 1-Wire**, при этом он имеет возможность только отправлять запросы на получение данных от ведомых устройств.

Для подключения контроллера CANNY 5 nano к шине 1-Wire может использоваться любой из его каналов ввода-вывода. При этом, данный канал должен быть снаружи подтянут к напряжению 5В резистором номиналом от 3 кОм до 7 кОм, либо необходимо использовать внутренний подтягивающий резистор, при его наличии у используемого канала (для каналов №№7,8,9,13,16,17).

В контроллерах CANNY 5 nano предусмотрена возможность обращения к конкретному устройству на шине 1-Wire по его адресу, что позволяет организовать работу контроллера с несколькими ведомыми устройствами по одному каналу. Кроме того, используя несколько каналов контроллера, возможно последовательное подключение к нескольким шинам 1-Wire.

Драйвер Dallas 1-Wire в своей работе использует ресурсы каналов контроллера, но имеет более высокий приоритет чем драйвера ввода-вывода. Таким образом, при активации драйвера Dallas 1-Wire, для задействованных в его работе каналов, изменение значений в связанных с ними регистрах драйвера ввода-вывода будет проигнорировано контроллером.

Ведомое устройство должно иметь постоянное, а не паразитное питание.

1.5.8 Параметры пользовательской конфигурации

Основная статья: **CANNY 5 nano, 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.

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.

1.5.9 Энергонезависимая память (ЭНП)

Основная статья: CANNY 5 nano, Энергонезависимая память (ЭНП)

Для исключения потери критически важной информации (состояния контроллера, состояния внешних устройств и т. п.) при сбросе питания, в контроллере CANNY 5 nano 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.

1.5.10 Драйвер пульта ИК ДУ

Основная статья: CANNY 5 nano, Драйвер пульта ИК ДУ

Контроллер CANNY 5 nano позволяет принимать и передавать команды инфракрасных пультов дистанционного управления (ИК ДУ) в широко распространенных форматах NEC и extended NEC. Работа драйвера возможна в трех режимах: только прием, только передача или прием/передача. Все каналы ввода-вывода CANNY 5 nano, поддерживают работу в режиме передачи данных пультов ИК ДУ, но лишь 4 из них (№№7, 8, 9, 13) могут быть использованы для приема ИК-сообщений, при этом тот или иной используемый драйвером канал контроллера, в каждый отдельный момент, может работать либо только на прием, либо только на передачу.

При передаче команд ИК ДУ, используемый для этого канал контроллера CANNY 5 nano генерирует только модулирующий сигнал. Для формирования пакетов импульсов контроллеру требуется наличие несущей частоты, источником которой может выступать как один из каналов ВЧ ШИМ CANNY 5 nano, так и внешний генератор ШИМ. Прием команд ИК ДУ требует наличия внешнего демодулятора, например TSOP1736 или аналогичного.

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 задействованных в его работе каналов, изменение значений в связанных с ними регистрах драйвера дискретного ввода-вывода будет проигнорировано контроллером.

1.5.11 Аналого-цифровой преобразователь (АЦП)

Основная статья: **CANNY 5 nano, Аналого-цифровой преобразователь (АЦП)**

Восемь из восемнадцати каналов ввода-вывода CANNY 5 nano, а именно каналы №№10, 11, 12, 13, 14, 15, 16, 17, могут быть использованы в качестве независимых друг от друга аналого-цифровых преобразователей. Кроме того, контроллер имеет отдельный АЦП встроенного термодиода.

Для активации драйвера АЦП на соответствующем канале необходимо передать значение, не равное «0», в соответствующий каналу «Регистр включения аналого-цифрового преобразователя канала №XX».

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.

Доступные пользователю каналы АЦП имеют разрешение 12 бит.

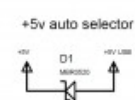
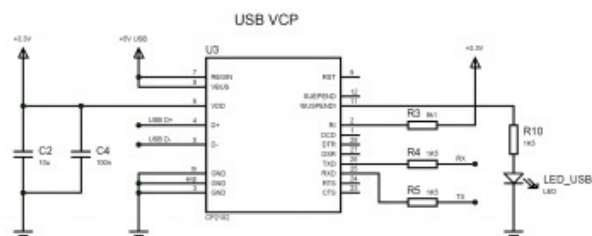
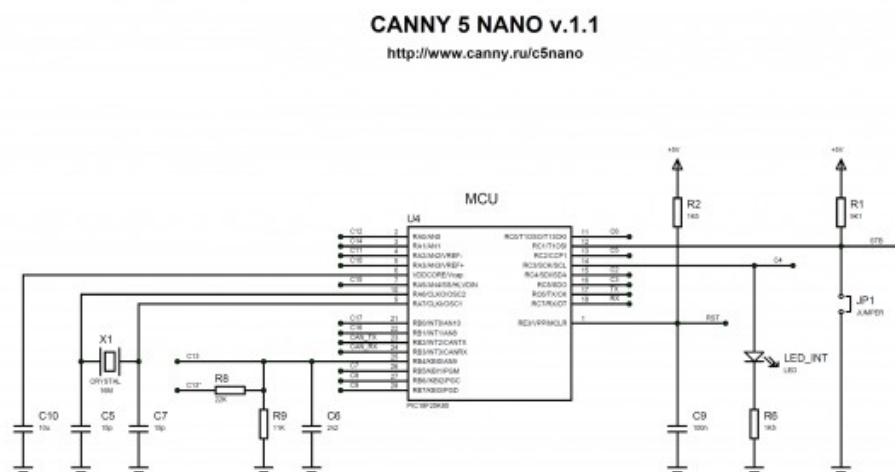
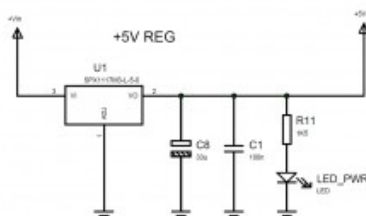
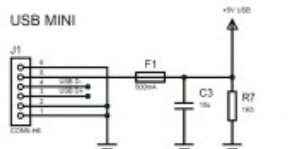
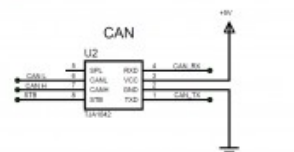
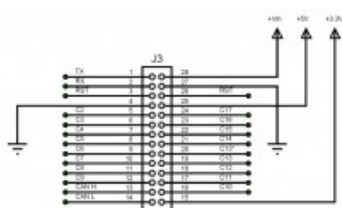
1.5.12 Драйвер датчика DHT11 / DHT22 / AM2302

Основная статья: **CANNY 5 nano, Драйвер датчика DHT11-DHT22-AM2302**

Контроллер CANNY 5 nano может работать с датчиками относительной влажности и температуры DHT11 / DHT22 / AM2302, используя однопроводное подключение с последовательным протоколом обмена данными, при этом контроллер выступает в качестве ведущего устройства, генерируя запросы к датчику, а DHT11 / DHT22 / AM2302 - в качестве ведомого, передавая данные замеров влажности и температуры, отвечая на данные запросы.

Для подключения контроллера CANNY 5 nano к DHT11 / DHT22 / AM2302 может использоваться любой из его каналов ввода-вывода, таким образом к контроллеру может быть одновременно подключено до 18 таких датчиков - по одному датчику на каждый канал контроллера, однако драйвер может единовременно работать только с одним датчиком, т.е. при работе с несколькими датчиками DHT11 / DHT22 / AM2302 необходимо организовывать их последовательный опрос.

1.6 Принципиальная схема CANNY 5 nano



2 CANNY 5 nano, 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 CANNY Lab через контекстное меню элементов диаграммы типа «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 и 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» в течении первого цикла выполнения функциональной диаграммы.

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	≥ 1	= запустить процедуру принудительного сброса контроллера
	0	= значение игнорируется

2.3 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 приблизительно 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. На практике, в силу особенностей подсчета интервалов времени в режиме экономии энергии, длительность фазы «сна» зависит от конкретного экземпляра микропроцессора и внешних условий, и может составлять от 1024 мс до 2000 мс и более.

Для более точного подсчета временных интервалов в состоянии «сна» предусмотрена возможность калибровки внутреннего таймера контроллера при переходе в режим пониженного энергопотребления.

Регистр	Ожидаемые значения	
Регистр установки режима пониженного энергопотребления	≥ 1	= перейти в режим пониженного энергопотребления
	0	= вернуться в режим нормального энергопотребления
Регистр включения замера длительности программного цикла в режиме пониженного энергопотребления	≥ 1	= включить режим замера длительности программного цикла при переходе в режим «сна», выполнить калибровку внутреннего таймера контроллера;
	0	= не использовать калибровку внутреннего таймера контроллера при переходе в режим «сна».

Примечание:

При включенном режиме замера длительности программного цикла в режиме пониженного энергопотребления подстройка внутреннего таймера будет выполняться при каждом входе контроллера в состояние «сна», в том числе после каждого автоматического выполнения одного цикла пользовательской диаграммы между этапами бездействия CANNY5 nano, находящегося в режиме пониженного энергопотребления. Рекомендуется выполнять калибровку таймера однократно, только при активации режима экономии энергии.

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 «C7» (канал №7), «C8» (канал №8), «C9» (канал №9), «C13» (канал №13), «C16» (канал №16), «C17» (канал №17), если они сконфигурированы для работы в активном режиме;
- при включенном из функциональной диаграммы драйвере CAN, UART1, UART2, Dallas 1-Wire, I²C, ИК при изменении электрического потенциала на соответствующем драйверу контакте контроллера.

Примечание: Информация об изменениях электрического потенциала на контактах «C7», «C8», «C9», «C13», «C16», «C17» контроллера, сконфигурированным работы в активном режиме, и изменениях электрического потенциала на контактах контроллера, соответствующих включенному драйверу CAN, UART и т.д., доступна пользователю через специальный регистр состояния контроллера - "Регистр контроля активности интерфейсов контроллера".

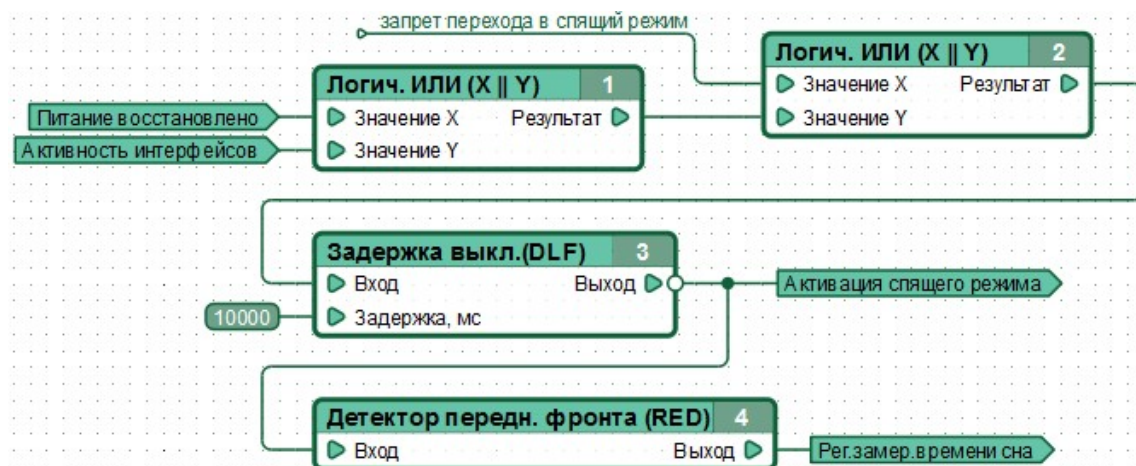
Информация о текущем режиме энергопотребления доступна при обращении к регистру «Регистр контроля режима пониженного энергопотребления».

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».

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.

2.4 Изменение разрешения системного таймера

Для расширения возможностей контроллера при работе с временными интервалами, например в качестве генератора импульсов, 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	≥ 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:



2.5 Фактическое время выполнения функциональной диаграммы

Время, требующееся контроллеру CANNY5 nano 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.

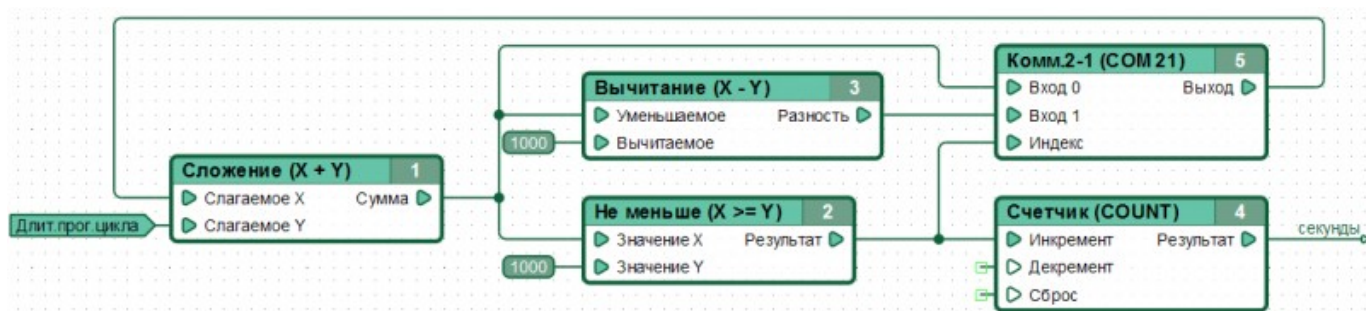
Информация о продолжительности предыдущего цикла выполнения функциональной диаграммы контроллера доступна в регистре «Регистр контроля длительности программного цикла».

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:



2.6 Идентификатор устройства

С выходом обновленного системного загрузчика контроллеров CANNY 5 nano 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.

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



2.7 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/ИК 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.

Регистр	Возвращаемые значения
Регистр контроля активности интерфейсов контроллера	1...65535 = 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

2.8 Идентификатор вендора устройства

С выходом обновленного системного загрузчика контроллеров CANNY 5 nano 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.

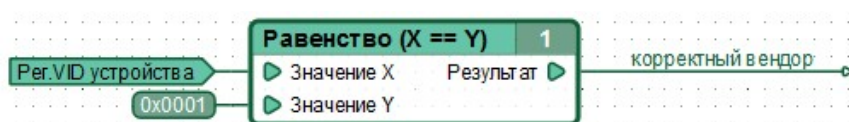
Идентификатор вендора устройства, назначаемый производителем контроллеров, является одинаковым для всех контроллеров, предназначенных для одного контрагента, или может быть установлен отдельно на конкретную партию контроллеров. Для его производства контроллеров с конкретным идентификатором вендора необходимо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

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

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.

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 CANNY 5 nano, Драйвер каналов ввода-вывода

3.1 Общее описание

Пользователям CANNY5 nano доступны восемнадцать дискретных каналов ввода-вывода общего назначения. Каждый канал физически представлен соответствующим контактом платы контроллера. Записывая данные в соответствующие регистры каналов драйвера, функциональная диаграмма может управлять электрическим потенциалом на каждом из этих контактов. Считывая данные регистров каналов ввода-вывода, диаграмма может получать информацию о текущем значении потенциала каждого из них.

Физические характеристики каналов позволяют подключать к ним светодиоды или слаботочные внешние управляющие устройства, например мощные транзисторы или транзисторные сборки, которые позволяют передавать управляющие сигналы на исполнительные устройства — электромагнитные реле, небольшие электродвигатели. В качестве внешних источников дискретных сигналов способных управлять работой контроллера, возможно использовать механические, электромеханические и электронные кнопки и переключатели, генераторы импульсов, источники напряжения 0-5В и 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.

Ряд драйверов контроллера CANNY 5 nano в своей работе используют ресурсы драйвера каналов ввода-вывода и, при этом, имеют более высокий приоритет. Таким образом, при использовании ими тех или иных каналов контроллера, доступ драйвера ввода-вывода к этим каналам невозможен.

ВНИМАНИЕ! Каналы CANNY5 nano не имеют защиты от короткого замыкания, перегрузки и перенапряжения. Во избежание выхода каналов из строя, избегайте превышения допустимых показателей напряжения и тока на каналах контроллера!

3.2 Особенности работы канала 13

Канал №13 может быть подключен через один из двух внешних контактов контроллера, обозначенных как «С13» и «*». При подключении через контакт «С13» канал №13, как и остальные контакты контроллера, может работать с сигналами напряжением 0-5В. При подключении через контакт «*» канал №13 подключается через резистивный делитель напряжения 22кОм / 11кОм , что дает возможность прямого включения данного контакта

в цепи напряжением от 0 до 15В.

3.3 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 конфигурации канала №0	≥ 1	= установить конфигурацию канала контроллера, определяющую текущий режим и параметры его работы, как «Дискретный положительный выход»;
Регистр конфигурации канала №1	0	= установить конфигурацию канала контроллера, определяющую текущий режим и параметры его работы, как «Дискретный положительный вход».
Регистр конфигурации канала №2		
...		
Регистр конфигурации канала №17		
Регистр включения подтягивающего резистора канала №7	≥ 1	= включить подтяжку канала контроллера к потенциалу +5В;
Регистр включения подтягивающего резистора канала №8	0	= отключить подтяжку канала контроллера к потенциалу +5В.
Регистр включения подтягивающего резистора канала №9		
Регистр включения подтягивающего резистора канала №13*		
Регистр включения подтягивающего резистора канала №16		
Регистр включения подтягивающего резистора канала №17		

<p>Регистр включения активного режима / счетчика передних фронтов канала №7</p> <p>Регистр включения активного режима / счетчика передних фронтов канала №8</p> <p>Регистр включения активного режима / счетчика передних фронтов канала №9</p> <p>Регистр включения активного режима / счетчика передних фронтов канала №13*</p> <p>Регистр включения активного режима / счетчика передних фронтов канала №16</p> <p>Регистр включения активного режима / счетчика передних фронтов канала №17</p>	<p>≥ 1</p> <p>0</p>	<p>= включить активный режим / счетчик передних фронтов канала контроллера;</p> <p>= отключить активный режим / счетчик передних фронтов канала контроллера.</p>
<p>Регистр включения активного режима / счетчика задних фронтов канала №7</p> <p>Регистр включения активного режима / счетчика задних фронтов канала №8</p> <p>Регистр включения активного режима / счетчика задних фронтов канала №9</p> <p>Регистр включения активного режима / счетчика задних фронтов канала №13*</p>	<p>≥ 1</p> <p>0</p>	<p>= включить активный режим / счетчик задних фронтов канала контроллера;</p> <p>= отключить активный режим / счетчик задних фронтов канала контроллера.</p>

Регистр включения активного режима / счетчика задних фронтов канала №16 Регистр включения активного режима / счетчика задних фронтов канала №17	
Регистр выходного значения канала №0 Регистр выходного значения канала №1 Регистр выходного значения канала №2 ... Регистр выходного значения канала №17	<div> <div>≥ 1</div> <div>= установить на соответствующем контакте контроллера электрический потенциал соответствующий состоянию «ВКЛ» (+5В);</div> </div> <div> <div>0</div> <div>= установить на соответствующем контакте контроллера электрический потенциал соответствующий состоянию «ВЫКЛ» (GND).</div> </div>

Примечание: При включении подтяжки или активного режима / счетчика фронтов канала №13 принимайте во внимание встроенную постоянную подтяжку данного канала к потенциалу GND за счет аппаратного делителя.

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 на каком-либо контакте контроллера, соответствующем каналам №№7, 8, 9 или 13, сконфигурированным как вход с включенным активным режимом, зарегистрировано изменение электрического потенциала;
	0	= за прошедший цикл выполнения функциональной диаграммы на каналах контроллера №№7, 8, 9 или 13, сконфигурированных как вход с включенным активным режимом, изменений не обнаружено.

Регистр входного значения канала №0 Регистр входного значения канала №1 ... Регистр входного значения канала №17	1 0	= на соответствующем контакте контроллера установлен электрический потенциал соответствующий полярности данного канала в режиме «ВКЛ» (+5В); = на соответствующем контакте контроллера установлен электрический потенциал не соответствующий полярности данного канала в режиме «ВКЛ» (+5В).
Регистр значения счетчика канала №7 Регистр значения счетчика канала №8 Регистр значения счетчика канала №9 Регистр значения счетчика канала №13 Регистр значения счетчика канала №16 Регистр значения счетчика канала №17	0...65535 = значение счетчика канала контроллера.	

3.4 Состояние канала в момент запуска контроллера

При включении контроллера, все его каналы находятся в режиме входа без внутренней подтяжки.

3.5 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, задается константой, определяющей электрический потенциал на контакте соответствующего канала контроллера, в положениях «ВКЛ» и «ВЫКЛ».

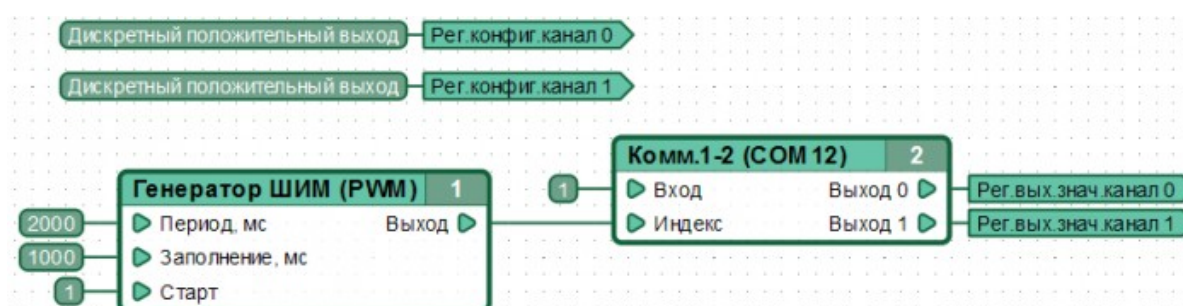
Параметр	Перечень допустимых значений
Тип канала	Дискретный выход

Состояние «ВКЛ»	+5В
Состояние «ВЫКЛ»	GND

Именованные константы конфигураций каналов, содержатся в разделе «Конфигурация канала 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 данному режиму работы.

Пример функциональной диаграммы выполняющей ежесекундное поочередное переключение электрических потенциалов с +5В на GND на контактах контроллера, соответствующих каналам №0 и №1.



ВНИМАНИЕ! Каналы CANNY5 папо не имеют защиты от короткого замыкания, перегрузки и перенапряжения. Во избежание выхода каналов из строя, избегайте превышения допустимых показателей напряжения и тока на каналах контроллера!

3.6 Режим 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 +5В, соответствующий состоянию «ВКЛ», и возвращает значение «0» в регистре входного значения, когда на соответствующем контакте контроллера установился электрический потенциал GND, не соответствующий состоянию «ВКЛ» канала.

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 №№7, 8, 9, 13, 16, 17 если конфигурацией установлен активный режим их работы, приведет к немедленному автоматическому выходу контроллера из режима пониженного энергопотребления при нахождении контроллера в нём. При этом канал работает с максимальной чувствительностью, т.е. «защита от дребезга» не предусмотрена и должна быть реализована, при необходимости, в функциональной диаграмме.

Пример функциональной диаграммы, устанавливающей состояние «ВКЛ» на контакте канала №1 контроллера при поступлении и удержании в течение не менее 200мс на соответствующем каналу №0 контакте контроллера, потенциала +5В:



3.7 Режим счетчика

При включении активного режима любого из каналов №№7, 8, 9, 13, 16, 17 контроллера, сконфигурированного как дискретный вход, CANNY5 nano позволяет асинхронно функциональной диаграмме производить автоматический подсчет импульсов входного сигнала накопительным итогом.

Асинхронный режим подсчета позволяет добиться большей точности подсчета количества поступивших импульсов и регистрировать сигналы большей частоты, чем при организации счетчика импульсов средствами функциональной диаграммы.

Сконфигурированный для работы в активном режиме / режиме счетчика канал возвращает в специальном регистре значений счетчика число импульсов, соответствующее: суммарному количеству переключений электрического потенциала на соответствующем контакте контроллера из состояния GND в состояние ≥ 2.5 вольт (≥ 7.5 для канала №13 при его подключении через контакт «*»), при включенном регистре счетчика передних фронтов; суммарному количеству переключений электрического потенциала на соответствующем контакте контроллера из состояния ≥ 2.5 вольт (≥ 7.5 для канала №13 при его подключении через контакт «*») в состояние GND, при включенном регистре счетчика задних фронтов; суммарному количеству передних и задних фронтов, при одновременно включенных соответствующих регистрах контроллера.

Примечание: При использовании канала №13 в режиме счетчика принимайте во внимание встроенную постоянную подтяжку данного канала к потенциалу

GND за счет аппаратного делителя.

Подсчет осуществляется накопительным итогом, при достижении значения 65535 и последующем увеличении на единицу, значение счетчика устанавливается равным нулю, каждый последующий импульс вновь увеличивает значение счетчика на единицу.

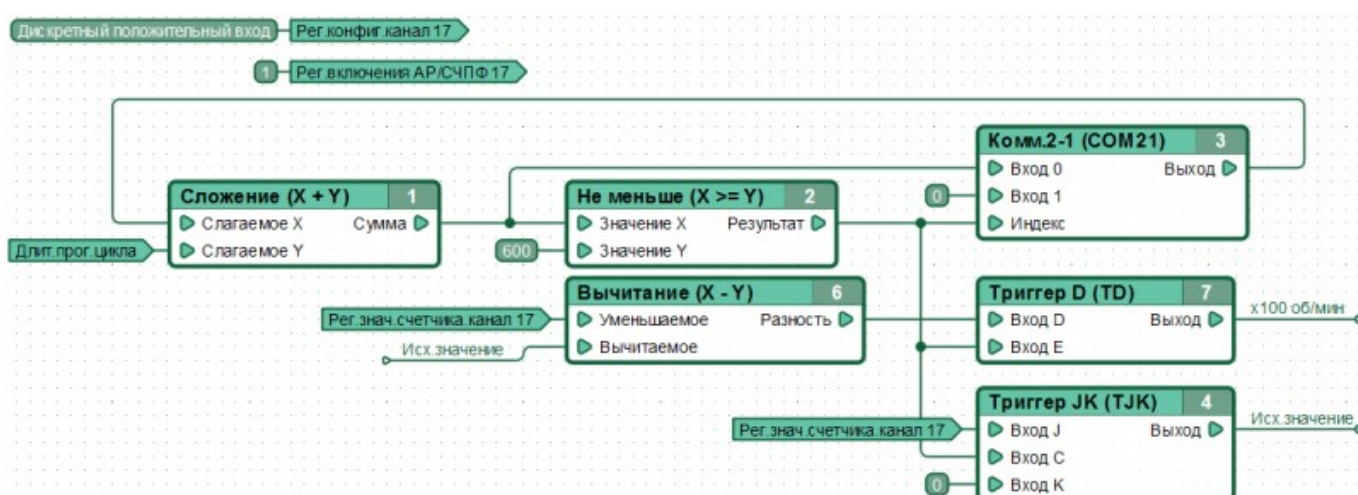
Для перевода канала контроллера в режим входа-счетчика, необходимо:

- в соответствующий каналу «Регистр конфигурации канала №XX» передать значение константы, соответствующей дискретному положительному входу;
- в соответствующий каналу «Регистр включения активного режима / счетчика передних фронтов канала №XX» или/и «Регистр включения активного режима / счетчика задних фронтов канала №XX» передать числовое значение, отличное от нуля.

Кроме того, при использовании каналов в активном режиме / режиме счетчик сохраняется возможность раздельного управления встроенной подтяжкой каждого из каналов через соответствующие каналам регистры «Регистр включения подтягивающего резистора канала №XX».

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

Пример функциональной диаграммы тахометра с разрешающей способностью 100 оборотов в минуту, подсчитывающем число импульсов на соответствующем канале №17 контакте контроллера CANNY5 nano:



Примечание: Работа канала CANNY5 папо в режиме входа-счетчика аналогична работе каналов контроллера CANNY7 в режиме счетчика с периодом равным нулю.

4 CANNY 5 nano, Драйвер широтно-импульсного модулятора (ШИМ)

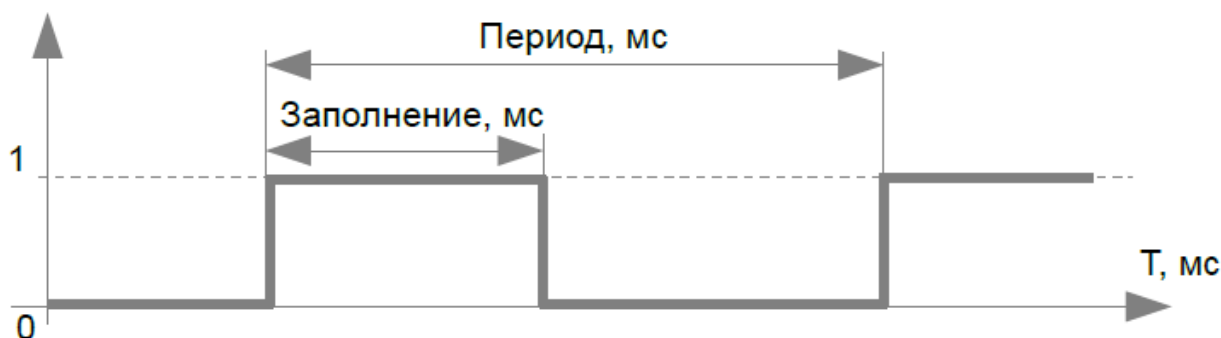
4.1 Общее описание

Три из восемнадцати каналов ввода-вывода (Канал №5, Канал №7 и Канал №16) CANNY5 nano поддерживают работу в режиме широтно-импульсного модулятора.

Использование драйвера ШИМ позволяет указанным каналам контроллера работать асинхронно функциональной диаграмме, что дает возможность добиться максимальной стабильности временных параметров генерируемого сигнала.

Драйвер ШИМ CANNY5 nano имеет два основных режима работы, задаваемых отдельно для каждого из трех каналов: низкочастотный и высокочастотный. Низкочастотный режим, позволяет реализовывать импульсный режим работы соответствующего канала с периодом от 2 до 65536 миллисекунд с шагом 1 миллисекунда, при использовании стандартных настроек системного таймера, и периодом от 200 до 6553600 микросекунд с шагом 100 микросекунд, при использовании увеличенного разрешения системного таймера.

В режиме высокочастотного ШИМ, период задается в диапазоне от 2 до 256 микросекунд, с шагом 1 микросекунда вне зависимости от настроек масштаба времени системного таймера.



Каналы могут быть задействованы независимо друг от друга и иметь индивидуальные настройки заполнения (скважности) ШИМ и подтяжки линии. В режиме низкочастотного ШИМ каналы также могут иметь и индивидуальные настройки значений периода, в то время как в режиме высокочастотного ШИМ, период является параметром, общим для всех использующих данный режим ШИМ каналов.

Каналы ШИМ имеют независимую во всех режимах работы настройку активного потенциала канала: открытый коллектор либо +5В.

Определение параметров работы канала для работы в режиме ШИМ, определяется записью ряда констант, в соответствующие регистры драйвера. Для конфигурации драйвера пользователю доступны несколько именованных констант.

Параметр	Перечень допустимых значений
Конфигурация высокочастотного режима ШИМ	Высокочастотный режим канала ШИМ включен, разрешение, 1мкс; Высокочастотный режим канала ШИМ выключен, разрешение, 1мс / 100мкс.
Конфигурация режима открытого коллектора ШИМ	Режим открытого коллектора канала ШИМ включен; Режим открытого коллектора канала ШИМ выключен.

Именованные константы, 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 широтно-импульсного выхода, необходимо:

- в «Регистр включения драйвера ШИМ канала №XX» передать числовое значение, отличное от «0»;
- в «Регистр включения высокочастотного режима ШИМ канала №XX» передать значение именованной константы, соответствующей выбранному режиму работы;
- в «Регистр установки режима открытого коллектора ШИМ канала №XX» передать значение именованной константы, соответствующей выбранному режиму работы;
- в «Регистр установки периода ШИМ канала №XX» передать числовое значение от 0 до 65565 (для низкочастотного ШИМ), либо числовое значение от 0 до 256 (для высокочастотного ШИМ), устанавливающее период генерируемых импульсов;
- в «Регистр установки скважности ШИМ канала №XX» передать числовое от 0 до 65565 (для низкочастотного ШИМ), либо числовое значение от 0 до 256 (для высокочастотного ШИМ), устанавливающее заполнение (скважность) генерируемых импульсов в мс.

4.2 Регистры драйвера

Ниже приведено описание допустимых значений регистров управления работой драйвера широтно-импульсного модулятора.

Для каждого из каналов, поддерживающих работу в режиме ШИМ, настройки параметров импульсных сигналов доступны с помощью индивидуальных, для каждого из них, регистров драйвера.

Регистр	Ожидаемые значения
---------	--------------------

Регистр установки периода ШИМ канала №XX	<p>В режиме низкочастотного ШИМ (мс / 0,1мс):</p> <p>1...65535 = задать период генератора ШИМ указанного канала.</p> <p>В режиме высокочастотного ШИМ (мкс):</p> <p>1...256 = задать период генератора ШИМ указанного канала. Значение превышающее 256 будет взято по модулю 256.</p>
Регистр установки заполнения (скважности) ШИМ канала №XX	<p>В режиме низкочастотного ШИМ (мс / 0,1мс):</p> <p>1...65535 = задать заполнение (скважность) ШИМ указанного канала.</p> <p>В режиме высокочастотного ШИМ (мкс):</p> <p>1...256 = задать заполнение (скважность) ШИМ указанного канала. Значение превышающее 256 будет взято по модулю 256.</p>
Регистр включения драйвера ШИМ канала №XX	<p>1...65535 = перевести канал контроллера в режим широтно-импульсного модулятора;</p> <p>0 = использовать канал контроллера в обычном режиме.</p>
Регистр установки режима открытого коллектора ШИМ канала №XX	<p>1...65535 = режим открытого коллектора включен, полярность генерации ШИМ для канала «воздух» / GND;</p> <p>0 = режим открытого коллектора выключен, полярность генерации ВЧ ШИМ для канала +5В / GND.</p>
Регистр включения высокочастотного режима ШИМ канала №XX	<p>1...65535 = перевести канал контроллера в режим высокочастотного широтно-импульсного модулятора;</p> <p>0 = использовать канал в режиме низкочастотного ШИМ.</p>

4.3 Особенности работы

Для корректной работы драйвера ШИМ в высокочастотном режиме, в регистрах установки периода ШИМ всех каналов работающих в высокочастотном режиме должны быть установлены одинаковые значения.

При включении драйвера ШИМ канала, в случае если значение его периода указано равным 0, то, вне зависимости от указанного значения заполнения (скважности), канал

ШИМ будет всегда находиться в состоянии GND. В случае если значение заполнения (скважности) указано равным 0, то, вне зависимости от других заданных параметров, канал ШИМ будет всегда находиться в состоянии GND.

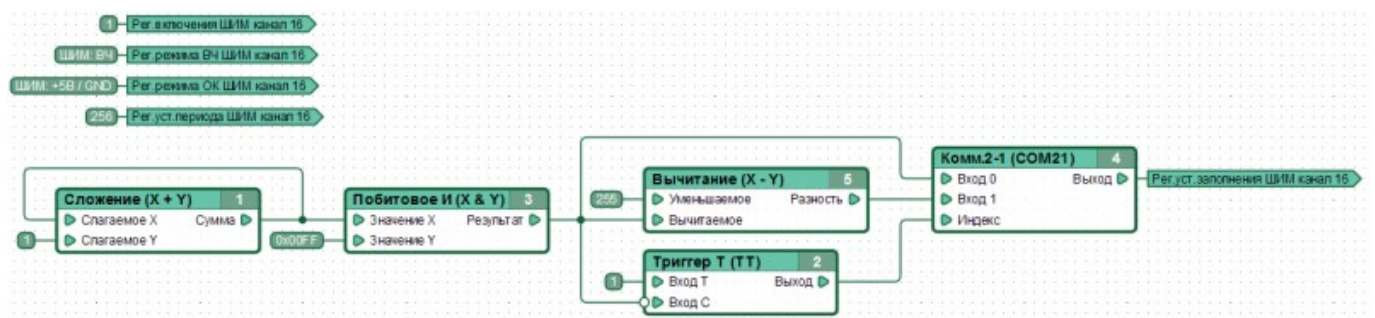
Если значение заполнения (скважности) больше значения периода или равно ему, то канал ШИМ будет всегда находиться в состоянии +5В или «воздух», в зависимости от значения регистра установки режима открытого коллектора ШИМ канала.

При использовании для работы с низкочастотным ШИМ режима увеличения разрешения системного таймера значения периода и заполнения задается в сотнях микросекунд, иначе — в миллисекундах.

При изменении значений периода или заполнения ШИМ во время работы канала в режиме ШИМ новые значения обновленных параметров будут применены драйвером только в начале следующего периода ШИМ. Для ускоренного применения новых параметров ШИМ очень низкой частоты, необходимо выполнять изменение значения через выключение режима ШИМ с последующим его включением.

Если канал включен в режиме ШИМ, т. е. в «Регистр включения драйвера ШИМ канала №XX» установлено значение, отличное от 0, то при установке в него значения 0 работа ШИМ канала немедленно прекращается и канал передается под управление драйвера ввода-вывода.

Функциональная диаграмма плавного изменения яркости светодиода, подключенного к каналу №16 контроллера:



ВНИМАНИЕ! Каналы CANNY5 nano не имеют защиты от короткого замыкания или перегрузки. Во избежание выхода каналов из строя, избегайте прямого подключения каналов с включенным выходным потенциалом GND к силовым цепям положительной полярности!

5 CANNY 5 nano, Драйвер UART

5.1 Общее описание

Контроллеры CANNY 5 nano имеют два независимых аппаратных интерфейса **UART** поддерживающих 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. Интерфейсы могут быть задействованы независимо друг от друга и иметь индивидуальные настройки скорости передачи данных и полярности сигналов. Интерфейс UART1 задействует каналы № 0 (TX) и № 1 (RX) контроллера, интерфейс UART2 задействует каналы № 8 (TX) и № 9 (RX). При включении интерфейса активируются оба его канала.

Каналы №0 и №1 контроллера параллельно подключены к USB-контроллеру виртуального последовательного порта, что позволяет, используя интерфейс UART1 организовать обмен данными между пользовательской диаграммой и ПК, либо другим USB-host устройством поддерживающим данный тип соединения.

Объем приемного и передающего буферов данных драйвера UART CANNY 5 nano составляет 32 байта.

*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.

5.2 Регистры драйвера

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

Регистры конфигурации драйвера UART.

Регистр	Ожидаемые значения
Регистр конфигурации UARTx	1...N = установить конфигурацию канала драйвера UART контроллера, определяющую текущий режим и

	<p>параметры его работы (задается специальной константой из справочника констант);</p> <p>0 = отключить канал от драйвера UART, вернуть управление каналом драйверу каналов ввода-вывода и разрешить изменения его состояния из функциональной диаграммы.</p>
Регистр установки таймаута приема сообщения UARTx, мс	<p>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;</p> <p>0 = использовать значение по умолчанию, задаваемое в конфигурации канала (13).</p>
Регистр включения режима открытого коллектора канала UARTx TX	<p>≥ 1 = отключить верхнее плечо канала передачи соответствующего интерфейса UART, передача будет осуществляться в режиме «воздух / GND»;</p> <p>0 = включить верхнее и нижнее плечо канала передачи соответствующего интерфейса UART передача будет осуществляться в режиме «+5В / GND».</p>

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; инверсная
Режим открытого коллкетора	"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.

Регистры диагностики драйвера UART.

Регистр	Возвращаемые значения
Регистр переполнения буфера UARTx	<p>1 = буфер UART переполнен;</p> <p>0 =</p>

		переполнения не зафиксировано.
Регистр отсутствия активности драйвера UARTx	1	= активность соответствующего канала драйвера UART отсутствует, линия находится в пассивном режиме;
	0	= зафиксирована активность на линии соответствующего канала драйвера UART.
Регистр ошибки приема UARTx	1	= во время приема данных UART произошла ошибка;
	0	= драйвер работает в нормальном режиме.
Регистр готовности буфера передачи данных UARTx	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 UARTx	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 UARTx	0...32	= значение, равное количеству байт данных, в принятом по соответствующему каналу драйвера UART пакете данных.
Регистр принятого сообщения UARTx D1:D0 ... Регистр принятого сообщения UARTx D31:D30	0...0xFFFF = значения соответствующих байт данных приемных буферов UART каждого канала, по два байта на регистр.	

Регистры передачи драйвера UART.

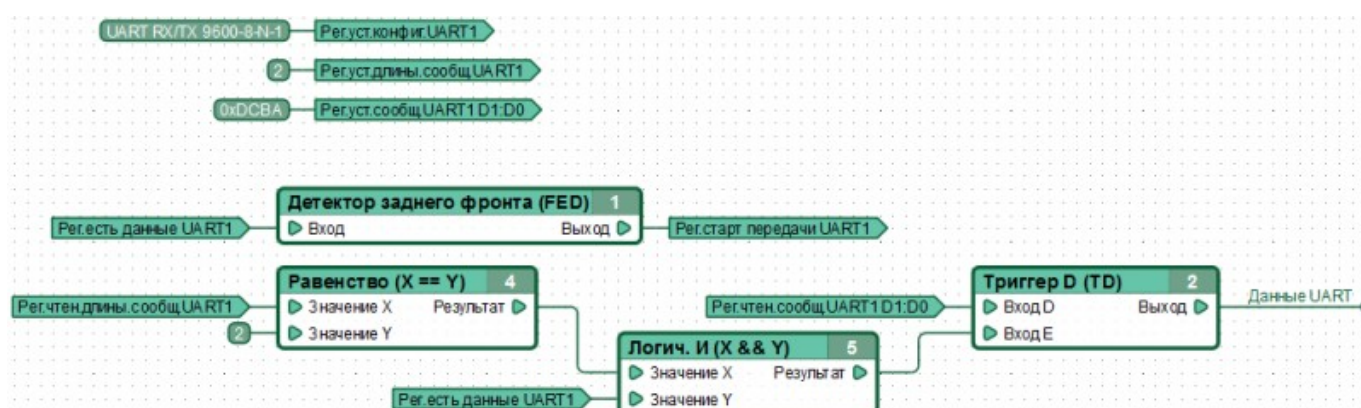
Регистр	Ожидаемые значения	
Регистр начала передачи UARTx	≥ 1	= загрузить данные из регистров передачи в буфер передачи соответствующего канала драйвера UART;
	0	= не загружать данные в буфер передачи соответствующего канала драйвера UART.
Регистр длины сообщения	0...32	= количество байт данных, которое будет

передачи UARTx	необходимо передать в линию, при получении команды на отправку соответствующего канала драйвера UART.
Регистр сообщения передачи UARTx D1:D0 ... Регистр сообщения передачи UARTx D31:D30	0...0xFFFF = значения соответствующих байт данных для передачи по соответствующему каналу драйвера UART, по два байта на регистр.

5.3 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.



6 CANNY 5 nano, Драйвер CAN

6.1 Общее описание

Два специальных контакта контроллера CANNY 5 nano, CAN-H и CAN-L, предназначены для подключения к цифровой информационной **шине CAN**.

Особенностью драйвера CAN контроллера CANNY 5 nano является то, что у него количество фильтров принимаемых сообщений CAN равно 8, против 16 у CANNY 7.

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:

Регистр	Ожидаемые значения
Регистр установки конфигурации CAN	1...N = активизация драйвера и установка скорости приема/передачи CAN-сообщений (задается специальной константой из справочника констант); 0 = драйвер отключен.
Регистр установки фильтра приема CAN IDL №0 ... Регистр установки фильтра приема CAN 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 CAN IDH №0 ... Регистр установки фильтра приема CAN 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 CAN	≥ 1	= включен режим пассивного приема (listen only) сообщений CAN;
	0	= включен режим нормального приема-передачи (normal) сообщений CAN;
Регистр режима фильтрации приема данных CAN	≥ 1	= режим фильтрации принимаемых сообщений CAN включен;
	0	= режим фильтрации принимаемых сообщений CAN отключен.
Регистр запрета автоматической повторной отправки сообщения CAN при ошибке	≥ 1	= режим запрета автоматической повторной отправки сообщения CAN при ошибке включен, драйвером будет предприниматься только одна попытка отправить данные CAN;
	0	= 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.

Параметр	Перечень допустимых значений
Скорость приема/передачи	20; 33; 50; 83; 95.2; 100; 125; 250; 500; 1000

данных, Кбит/с	
----------------	--

Именованные константы, определяющие конфигурацию CAN-драйвера, содержатся в разделе «Конфигурация CAN» справочника констант CanpyLab, 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 CAN	1	= активность драйвера CAN отсутствует, шина бездействует, прием данных не осуществляется;
	0	= регистрируется активность CAN.
Регистр переполнения буфера приема CAN	1	= ошибка, буфер CAN переполнен;
	0	= переполнение буфера приема отсутствует.
Регистр ошибки приема/передачи CAN	1	= уровень ошибок приема/передачи CAN превысил допустимый порог;
	0	= уровень ошибок приема/передачи CAN ниже допустимого порога.
Регистр готовности буфера передачи данных CAN	1	= буфер передачи данных драйвера CAN свободен и готов к загрузке новых сообщений;
	0	= буфер передачи данных драйвера CAN не готов.

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

Регистр	Возвращаемые значения	
Регистр наличия принятых данных CAN	1	= 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;
	0	= в буфере приема драйвера CAN нет актуальных данных.
Регистр принятого сообщения CAN IDL	0...0xFFFF = значение младшей части идентификатора полученного CAN-сообщения.	
Регистр принятого сообщения CAN IDH	0...0x1FFF = значение старшей части идентификатора полученного CAN-сообщения.	
	0...0xC000 =	

Регистр принятого сообщения CAN ERL	значение, равное количеству байт данных в принятом сообщении, признаки EXT и RTR (см. примечание).
Регистр принятого сообщения CAN D1:D0 ... Регистр принятого сообщения CAN D7:D6	0...0xFFFF = 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.

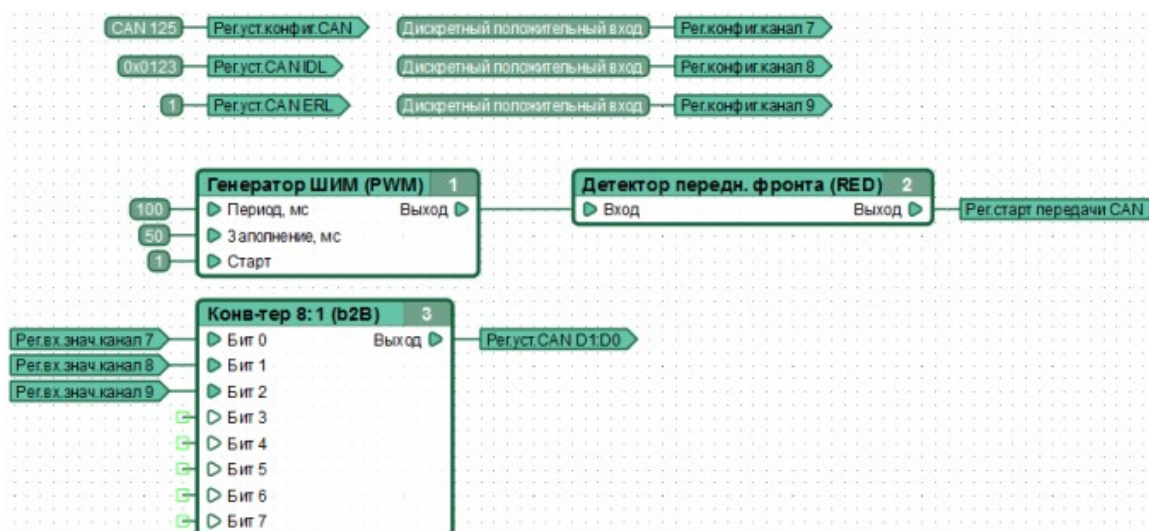
Регистры передачи сообщений CAN используются для размещения в буфере передачи драйвера данных, подлежащих отправке.

Регистр	Ожидаемые значения
Регистр начала передачи CAN	≥ 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	0...0xFFFF = значение младшей части идентификатора передаваемого CAN-сообщения.
Регистр сообщения передачи CAN IDH	0...0x1FFF = значение старшей части идентификатора передаваемого CAN-сообщения.
Регистр сообщения передачи CAN ERL	0...0xC000 = значение, равное количеству байт данных в передаваемом сообщении, признаки EXT и RTR (см. примечание).
Регистр сообщения передачи CAN D1:D0 ... Регистр сообщения передачи CAN D7:D6	0...0xFFFF = значения соответствующих байт данных передаваемого сообщения CAN, по два байта на регистр.

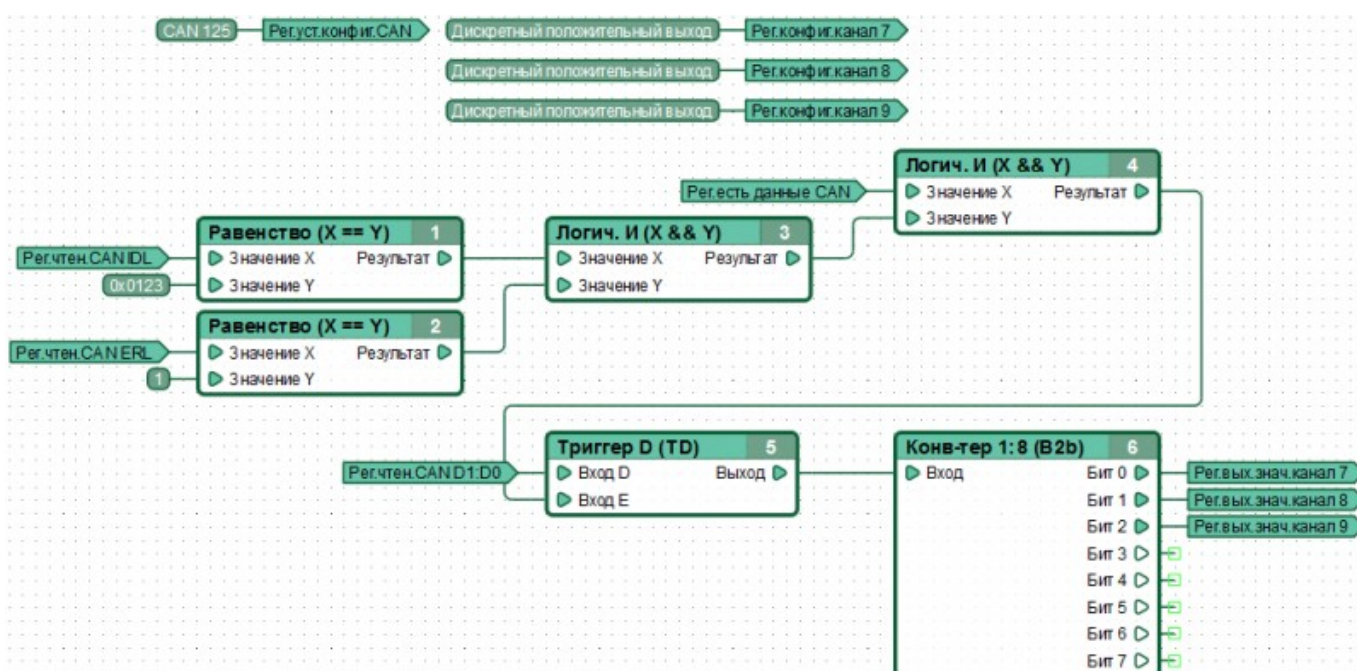
Примечание: Регистр сообщения передачи CAN ERL, помимо числа байт в передаваемом сообщении 0...8 в младших битах, содержит в своих старших битах информацию о специальных признаках сообщения: бит 15 - признак EXT и бит 14 признак RTR. Где EXT = 1 при передаче сообщения в расширенном

формате, $EXT = 0$ при стандартном формате сообщения; $RTR = 1$ при передаче удаленного запроса данных, $EXT = 0$ при передаче обычного сообщения.

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.



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.



7 CANNY 5 nano, Драйвер I²C

7.1 Общее описание

Два из восемнадцати каналов ввода-вывода CANNY 5 nano, а именно каналы №2 и №4, поддерживают работу в режиме приема/передачи данных, с использованием протокола I²C, и могут быть использованы для связи контроллеров с внешним оборудованием поддерживающим данный протокол связи.

Контроллер CANNY5 nano может выступать только в качестве ведущего узла сети, при этом он имеет возможность как передавать данные ведомым устройствам, так и отправлять запросы на получение данных от них. Обмен данными между устройствами происходит отдельными сеансами, с максимальной длиной сообщения I²C внутри одного сеанса равной 32 байтам, т. е. открытие одновременно несколько сеансов с разными устройствами не допускается. Скорость обмена фиксированная и составляет 100 кбит/с. Общее число ведомых устройств на линии может достигать нескольких десятков.

Драйвер I²C в своей работе использует ресурсы каналов контроллера, но имеет более высокий приоритет чем драйвер дискретного ввода-вывода. Таким образом, при активации драйвера I²C, для задействованных в его работе каналов, изменение значений в связанных с ними регистрах драйвера дискретного ввода-вывода будет проигнорировано контроллером.

Каналы №№2 и 4 активируются совместно, одновременно с установкой конфигурации драйвера I²C контроллера, т. е. его активацией. Канал №2 используется в качестве линии данных (SDA), канал №4 — как тактирующий канал (SCL).

Примечание: Активация драйвера I²C автоматически переводит каналы №2 и №4 в режим передачи/приема данных, делая невозможным их использование в качестве дискретных входов или выходов драйвера ввода-вывода, т. к. драйвер I²C имеет более высокий приоритет.

7.2 Регистры драйвера I²C

Ниже приведено описание допустимых значений регистров управления работой драйвера I²C.

Регистры конфигурации драйвера I²C.

Регистр	Ожидаемые значения
Регистр адреса I ² C	2...254 =

		четное число, адрес slave-устройства, с которым будет производиться обмен данными по шине I ² C.
Регистр активации драйвера I ² C	≥ 1	= активировать драйвер I ² C, передать ему управление каналами контроллера, используемыми в качестве линий SDA и SCL;
	0	= деактивировать драйвер I ² C, вернуть управление каналами драйверу ввода-вывода и разрешить изменения их состояний из функциональной диаграммы.

Регистры диагностики драйвера I²C.

Регистр	Возвращаемые значения	
Регистр ошибки приема/передачи данных I ² C	1	= во время приема или отправки сообщения I ² C произошла ошибка;
	0	= драйвер работает в нормальном режиме.
Регистр переполнения буфера передачи I ² C	1	= ошибка, буфер I ² C переполнен;
	0	= переполнение буфера приема отсутствует.
Регистр готовности буфера передачи данных I ² C	1	= буфер передачи данных драйвера I ² C свободен и готов к загрузке новых данных;
	0	= буфер передачи данных драйвера I ² C не готов.

Регистры приема драйвера I²C.

Регистр	Возвращаемые значения	
Регистр наличия принятых данных I ² C	1	= сообщение успешно получено и доступно в регистрах буфера приема драйвера I ² C;
	0	= в буфере приема драйвера I ² C отсутствуют актуальные данные.
Регистр длины принимаемого сообщения I ² C	0...32	= значение, равное количеству байт, которые должны быть приняты в сообщении I ² C.
Регистр принятых данных I ² C D1:D0 ... Регистр принятых данных I ² C D31:D30	0...0xFFFF = значения соответствующих байт данных приемного буфера I ² C, по два байта на регистр.	

Регистры передачи драйвера I²C.

Регистр	Ожидаемые значения
---------	--------------------

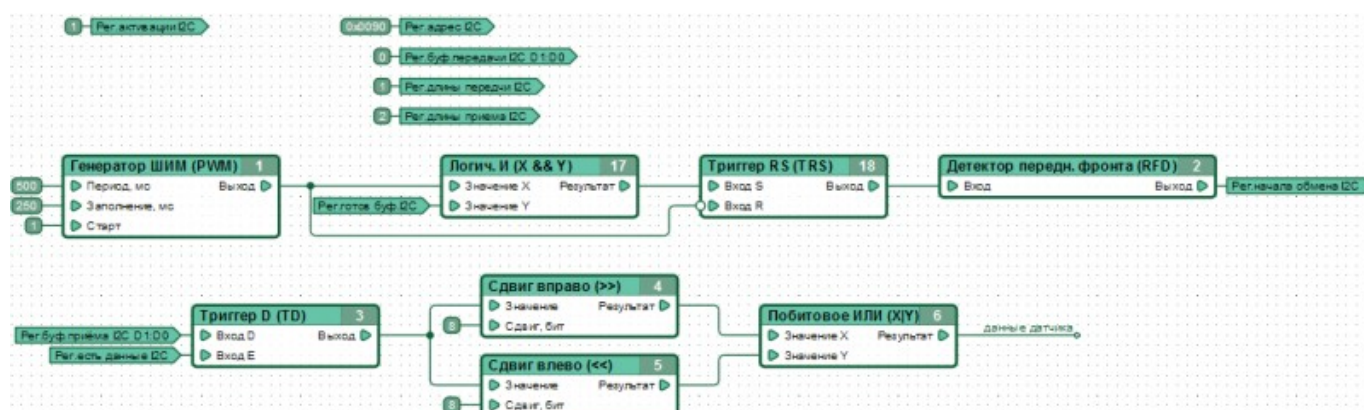
Регистр начала обмена данными I ² C	≥ 1 = загрузить данные из регистров передачи в буфер передачи / загрузить данные из буфера приема в регистры принятых данных драйвера I ² C; 0 = не загружать данные в буфер передачи / не считывать данные из буфера приема драйвера I ² C.
Регистр длины передаваемого сообщения I ² C	0...32 = количество байт сообщения I ² C, которое будет необходимо передать при получении команды на отправку данных.
Регистр передаваемого сообщения I ² C D1:D0 ... Регистр передаваемого сообщения I ² C D31:D30	0...0xFFFF = значения соответствующих байт сообщения I ² C для передачи, по два байта на регистр.

7.3 Особенности работы драйвера I²C

Обмен данными в сети I²C с использованием контроллера CANNY 5 nano определяется комбинацией значений регистров длины передаваемого и принимаемого сообщения, установленных пользователем (смотри таблицу).

Значение регистра длины передаваемого сообщения I ² C	Значение регистра длины принимаемого сообщения I ² C	Направление обмена данными
> 0	= 0	Только передача данных от CANNY7 (Master) ведомому (Slave) устройству с адресом, указанным в соответствующем регистре.
> 0	> 0	Передача данных от CANNY7 (Master) ведомому (Slave) устройству с адресом, указанным в соответствующем регистре, и получение от него ответных данных.
= 0	> 0	Только прием данных CANNY7 (Master) от ведомого (Slave) устройства с адресом, указанным в соответствующем

Пример функциональной диаграммы получения данных от датчика температуры TCN75. В процессе работы контроллер выступает в качестве ведущего (Master) узла шины I²C, выполняя опрос датчика температуры 2 раза в секунду. В этом примере CANNY5 nano (Master) выступает в качестве приемника, запрашивающего данные, а термодатчик (Slave) — в качестве передатчика. Такое направление передачи данных задается записью в регистры длины передаваемого сообщения I²C (Рег.длины передачи I²C) и длины принимаемого сообщения I²C (Рег.длины приема I²C) значений отличных от «0», соответствующих количеству передаваемых (запроса температуры) и получаемых (значения температуры) данных.



Каждые 500мс в шину, если буфер передачи CANNY5 nano свободен и готов к приему нового сообщения, ведомому узлу с адресом 0x90, передается запрос на получение значения температуры. В ответ предполагается получение 2 байт данных.

При получении данных от термодатчика выполняется их предварительная обработка, результаты которой сохраняются в именованной сети «данные датчика».

ВНИМАНИЕ! Канал 4 CANNY5 nano параллельно используется для управления встроенным контрольным светодиодом (LED — смотри принципиальную электрическую схему контроллера), в следствие чего, его работа в режиме I²C требует дополнительной подтяжки к напряжению +5В с помощью внешнего резистора, номиналом 400...600 Ом, либо удаления с платы CANNY5 nano контрольного светодиода.

8 CANNY 5 nano, Драйвер Dallas 1-Wire

8.1 Общее описание

Контроллер CANNY 5 nano может быть использован в качестве ведущего (MASTER) узла в однопроводной сети передачи данных **Dallas 1-Wire**, при этом он имеет возможность только отправлять запросы на получение данных от ведомых устройств.

Для подключения контроллера CANNY 5 nano к шине 1-Wire может использоваться любой из его каналов ввода-вывода. При этом, данный канал должен быть снаружи подтянут к напряжению 5В резистором номиналом от 3 кОм до 7 кОм, либо необходимо использовать внутренний подтягивающий резистор, при его наличии у используемого канала (для каналов №№7,8,9,13,16,17).

В контроллерах CANNY 5 nano предусмотрена возможность обращения к конкретному устройству на шине 1-Wire по его адресу, что позволяет организовать работу контроллера с несколькими ведомыми устройствами по одному каналу. Кроме того, используя несколько каналов контроллера, возможно последовательное подключение к нескольким шинам 1-Wire.

Драйвер Dallas 1-Wire в своей работе использует ресурсы каналов контроллера, но имеет более высокий приоритет чем драйвера ввода-вывода. Таким образом, при активации драйвера Dallas 1-Wire, для задействованных в его работе каналов, изменение значений в связанных с ними регистрах драйвера ввода-вывода будет проигнорировано контроллером.

Ведомое устройство должно иметь постоянное, а не паразитное питание.

8.2 Регистры драйвера 1-Wire

Ниже приведено описание регистров управления работой драйвера 1-Wire.

Регистры конфигурации драйвера 1-Wire.

Для активации драйвера 1-Wire необходимо установить номер канала контроллера, подключенного к шине 1-Wire, с которым предполагается начать работу, в соответствующий регистр драйвера.

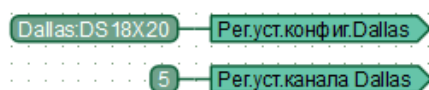
Примечание: В каждый отдельный момент времени драйвер может работать только с одной из подключенных к нему шин 1-Wire, при этом допускается динамическое переключение между шинами в процессе выполнения

функциональной диаграммы.

Управление отправкой запросов на получение данных от ведомых устройств и контроль получения их ответов (опросов датчиков) выполняется с помощью соответствующих регистров драйвера 1-Wire.

Примечание: Во избежание потери данных при работе с несколькими шинами Dallas, выполняйте переключение между ними только после получения от ведомых устройств ответов на запрос контроллера и обработки или сохранения полученных данных.

Пример конфигурации канала №5 для работы с датчиком температуры DS18X20.



Регистры конфигурации драйвера 1-Wire.

Регистр	Ожидаемые значения
Регистр конфигурации Dallas	<p>1...N = активизация драйвера и установка типа подключаемого устройства (задается специальной константой из справочника констант);</p> <p>0 = отключить канал от драйвера 1-Wire, вернуть управление каналом драйверу каналов ввода-вывода и разрешить изменения его состояния из функциональной диаграммы.</p>
Регистр номера канала Dallas	0...17 = установить номер канала контроллера используемого для работы с шиной 1-Wire.

Конфигурация канала для работы в данном режиме задается следующим именованными константами:

Параметр	Перечень допустимых значений
Тип устройства 1-Wire	Датчик температуры DS18X20, Считыватель CP-Z в режиме эмуляции DS1990A, Считыватель DS1990A

Регистры передачи драйвера 1-Wire.

С помощью регистров передачи драйвера 1-Wire пользователь может задать адрес опрашиваемого ведомого устройства и отправить запрос на получение данных.

Регистр	Ожидаемые значения
---------	--------------------

Регистр начала передачи Dallas	≥ 1 = отправить ведомому устройству запрос на получение данных; 0 = не отправлять ведомому устройству запрос на получение данных.
Регистр установки 64-битного ROM-кода адресата передачи Dallas SN0:FC	0...0xFFFF = значение регистра: младшая часть содержит идентификатор семейства устройства (FC), старшая часть содержит первый байт уникального адреса устройства (SN0).
Регистр установки 64-битного ROM-кода адресата передачи Dallas SN2:SN1	0...0xFFFF = значение регистра: младшая часть содержит второй байт уникального адреса устройства (SN1), старшая часть содержит третий байт уникального адреса устройства (SN2).
Регистр установки 64-битного ROM-кода адресата передачи Dallas SN4:SN3	0...0xFFFF = значение регистра: младшая часть содержит четвертый байт уникального адреса устройства (SN3), старшая часть содержит пятый байт уникального адреса устройства (SN4).
Регистр установки 64-битного ROM-кода адресата передачи Dallas CRC:SN5	0...0xFFFF = значение регистра: младшая часть содержит шестой байт уникального адреса устройства (SN5), старшая часть содержит контрольную сумму ROM-кода адресата (CRC).

Примечание: Регистры установки 64-битного ROM-кода адресата передачи Dallas используются только для выбора устройства на шине 1-Wire по его уникальному номеру (ROM-коду), в случае подключения к одной шине нескольких устройств. При работе с единственным устройством на шине, в случае если его ROM-код неизвестен, установите значение «0» во все регистры ROM-кода адресата передачи Dallas.

Регистры приема драйвера 1-Wire.

В процессе работы, при получении от ведомых устройств ответов на запросы контроллера, данные, в зависимости от конфигурации драйвера 1-Wire, размещаются в его соответствующие регистры приема.

Общие регистры приема.

Регистр	Возвращаемые значения
Регистр наличия принятых данных Dallas	1 = данные от ведомого устройства успешно получены и доступны в регистрах чтения драйвера Dallas; 0 = в буфере приема драйвера отсутствуют актуальные данные.

Прием в режиме CZP / DS1990A.

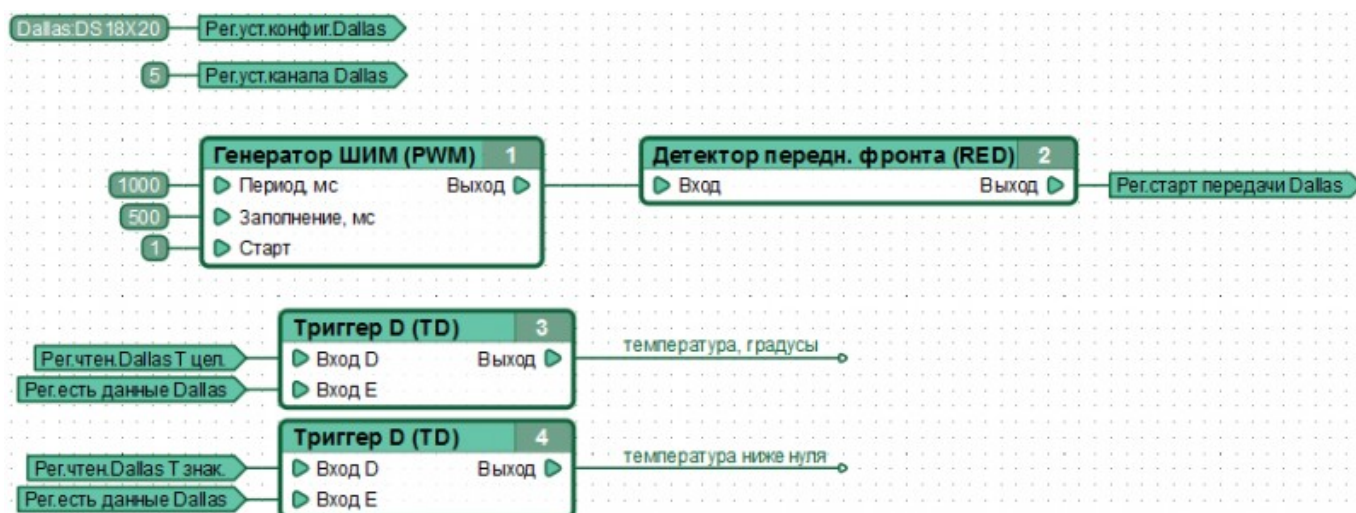
Регистр	Возвращаемые значения
Регистр чтения 64-битного ROM-кода устройства Dallas SN0:FC	0...0xFFFF = значение регистра: младшая часть содержит идентификатор семейства ключа (FC), старшая часть содержит первый байт уникального адреса ключа (SN0).
Регистр чтения 64-битного ROM-кода устройства Dallas SN2:SN1	0...0xFFFF = значение регистра: младшая часть содержит второй байт уникального адреса ключа (SN1), старшая часть содержит третий байт уникального адреса ключа (SN2).
Регистр чтения 64-битного ROM-кода устройства Dallas SN4:SN3	0...0xFFFF = значение регистра: младшая часть содержит четвертый байт уникального адреса ключа (SN3), старшая часть содержит пятый байт уникального адреса ключа (SN4).
Регистр чтения 64-битного ROM-кода устройства Dallas CRC:SN5	0...0xFFFF = значение регистра: младшая часть содержит шестой байт уникального адреса ключа (SN5), старшая часть содержит контрольную сумму ROM-кода ключа (CRC).

Прием в режиме DS18X20.

Регистр	Возвращаемые значения
Регистр чтения показаний температуры: целые градусы Цельсия	0...125 = модуль целой части значения измеренной датчиком температуры.
Регистр чтения показаний температуры: десятитысячные доли градуса Цельсия	0...9999 = модуль десятичной части значения измеренной датчиком температуры.
Регистр чтения показаний температуры: знак (0 = выше нуля; 1 = ниже нуля)	0 = измеренная датчиком температура не ниже нуля; 1 = измеренная датчиком температура ниже нуля.

Примечание: При приеме в режиме DS18X20 период опроса датчика температуры, т. е. отправки ему запросов с помощью регистра начала передачи Dallas, должен быть не менее 750мс.

Пример функциональной диаграммы работы с термодатчиком DS18X20.



Опрос термодатчика, подключенного к каналу №5 контроллера, осуществляется 1 раз в секунду. При получении данных от DS18X20, в регистре наличия принятых данных драйвера Dallas появляется значение «1» и данные из регистров чтения показаний температуры, с помощью D-триггеров, сохраняются в соответствующие именованные сети для дальнейшей обработки.

9 CANNY 5 nano, Параметры пользовательской конфигурации

9.1 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.

9.2 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.

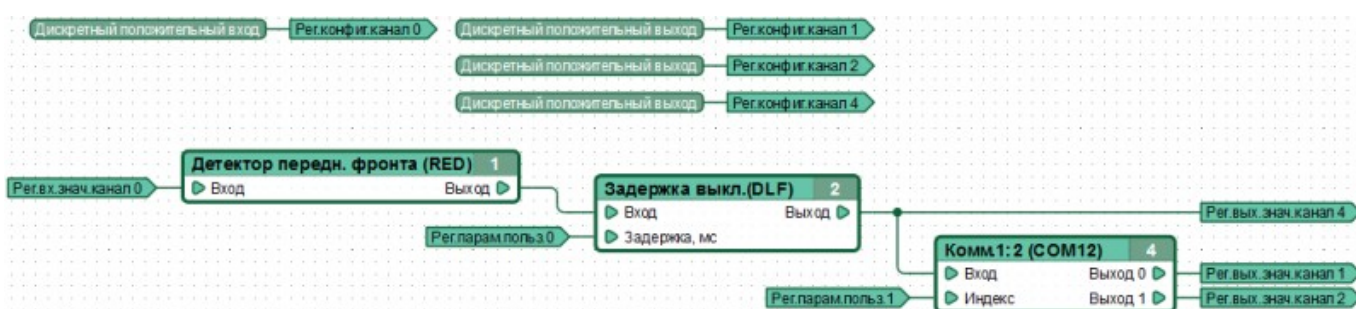
Значения в данных регистрах установятся при старте контроллера, после записи в него ПО посредством исполняемого файла автономной загрузки, и будут оставаться неизменными (константными) на протяжении всего времени работы функциональной диаграммы, не изменяясь даже при сбросе контроллера. Изменить значения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

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

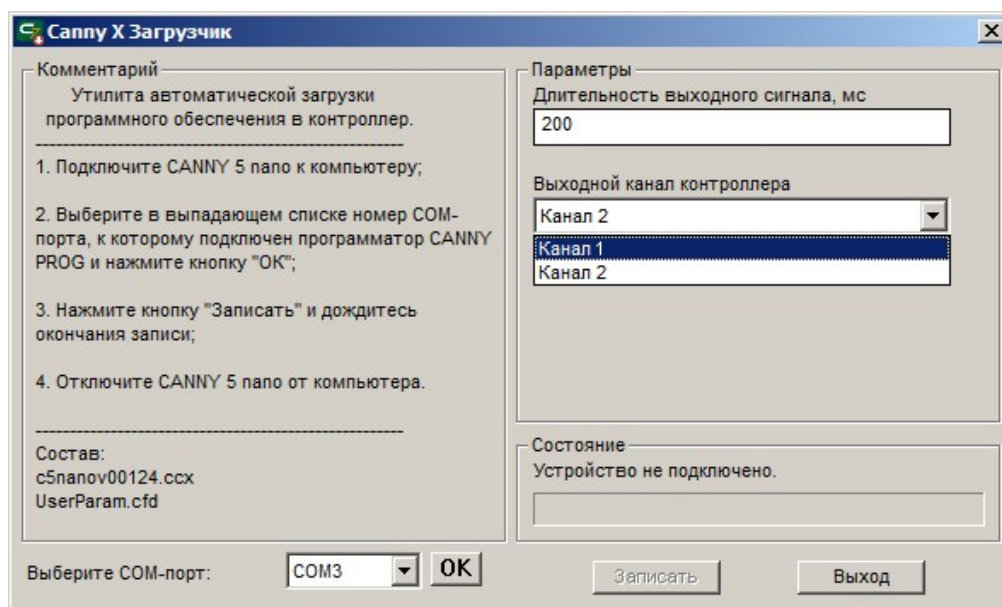
Примечание: При записи контроллера из среды CannyLab значения всех регистров параметров пользовательской конфигурации устанавливаются равным нулю.

9.3 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 «Выходной канал контроллера», Тип «Список». В список значений параметра «Выходной канал контроллера» добавьте две строки: Название «Канал 1», Значение «0» и «Канал 2», 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 №0 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.

10 CANNY 5 nano, Энергонезависимая память (ЭНП)

10.1 Общее описание

Для исключения потери критически важной информации (состояния контроллера, состояния внешних устройств и т. п.) при сбросе питания, в контроллере CANNY 5 nano 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.

10.2 Регистры энергонезависимой памяти

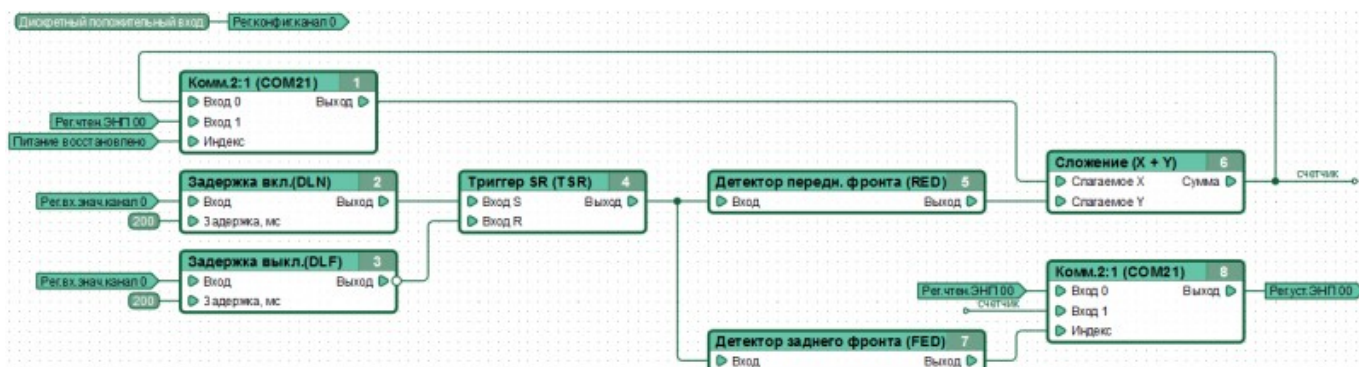
Ниже приведено описание допустимых значений регистров установки энергонезависимой памяти контроллера. Они используются для сохранения информации в ячейках ЭНП.

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 №0 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. Блоки №№2...4 применены для защиты от дребезга при нажатии на кнопку.

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 папо 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).

11 CANNY 5 nano, Драйвер пульта ИК ДУ

11.1 Общее описание

Контроллер CANNY 5 nano позволяет принимать и передавать команды инфракрасных пультов дистанционного управления (ИК ДУ) в широко распространенных форматах NEC и extended NEC. Работа драйвера возможна в трех режимах: только прием, только передача или прием/передача. Все каналы ввода-вывода CANNY 5 nano, поддерживают работу в режиме передачи данных пультов ИК ДУ, но лишь 3 из них (№№7, 8, 9) могут быть использованы для приема ИК-сообщений, при этом тот или иной используемый драйвером канал контроллера, в каждый отдельный момент, может работать либо только на прием, либо только на передачу.

При передаче команд ИК ДУ, используемый для этого канал контроллера CANNY 5 nano генерирует только модулирующий сигнал. Для формирования пакетов импульсов контроллеру требуется наличие несущей частоты, источником которой может выступать как один из каналов ВЧ ШИМ CANNY 5 nano, так и внешний генератор ШИМ. Прием команд ИК ДУ требует наличия внешнего демодулятора, например TSOP1736 или аналогичного.

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 задействованных в его работе каналов, изменение значений в связанных с ними регистрах драйвера дискретного ввода-вывода будет проигнорировано контроллером.

11.2 Регистры драйвера пульта ИК ДУ

Драйвер пульта ИК ДУ предусматривает возможность организации одновременного приема и передачи данных по двум независимым каналам. Кроме того, возможно подключение к CANNY 5 nano нескольких приемников/передатчиков ИК-сигналов.

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

Регистры конфигурации драйвера пульта ИК ДУ.

Регистр	Ожидаемые значения
Регистр конфигурации ИК-порта	1...N = установить конфигурацию канала драйвера пульта ИК ДУ контроллера, определяющую

	0	текущий режим и параметры его работы (задается специальной константой из справочника констант); = отключить канал от драйвера пульта ИК ДУ, вернуть управление каналом драйверу каналов ввода-вывода и разрешить изменения его состояния из функциональной диаграммы.
Регистр номера канала приема ИК-порта	7,8,9	= установить номер канала контроллера, используемый драйвером пульта ИК ДУ для приема данных.
Регистр номера канала передачи ИК-порта	0...17	= установить номер канала контроллера, используемый драйвером пульта ИК ДУ для передачи данных.

Примечание: В конфигурации «Прием и передача» для приема и передачи данных должны быть назначены отдельные каналы контроллера. Будьте внимательны: использовать для работы в данном режиме один и тот же канал контроллера нельзя!

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.

Параметр	Перечень допустимых значений
Стандарт	NEC
Направление потока данных	Прием, передача, прием и передача
Тип сигнала	Прямой («1»: плюс, «0»: открытый коллектор), инверсный («1»: минус, «0»: открытый коллектор).

Примечание: Для корректной работы контроллера в режиме ИК-приемника/передатчика требуется указание дополнительных настроек задействованных для этого каналов контроллера: для канала-приемника необходимо установить активный режим его работы и включить на нем подтягивающий резистор; для передатчика — включить подтягивающий резистор.

Пример конфигурации драйвера пульта ИК ДУ контроллера для работы в качестве приемника/передатчика ИК-сигналов, при этом для приема данных используется канал №7, а для передачи - канал №17.



Регистры диагностики драйвера пульта ИК ДУ.

Регистр	Возвращаемые значения	
Регистр отсутствия активности драйвера ИК	1	= активность драйвера ИК ДУ на соответствующем канале отсутствует;
	0	= зафиксирована активность драйвера ИК ДУ на соответствующем канале.
Регистр переполнения буфера драйвера ИК	1	= буфер соответствующего канала драйвера пульта ИК ДУ переполнен;
	0	= переполнения соответствующего канала драйвера пульта ИК ДУ не зафиксировано.
Регистр готовности буфера передачи ИК	1	= буфер передачи данных драйвера пульта ИК ДУ свободен и готов к работе;
	0	= буфер передачи данных драйвера пульта ИК ДУ занят.

Регистры приема драйвера пульта ИК ДУ.

Регистр	Возвращаемые значения	
Регистр наличия принятых данных ИК	1	= сообщение успешно получено и доступно в регистрах буфера приема драйвера пульта ИК ДУ;
	0	= в буфере приема драйвера пульта ИК ДУ отсутствуют актуальные данные.
Регистр принятого сообщения ИК D1:D0 Регистр принятого сообщения ИК D3:D2	0...0xFFFF = значения соответствующих байт данных приемного буфера драйвера пульта ИК ДУ, по два байта на регистр.	

Регистры передачи драйвера пульта ИК ДУ.

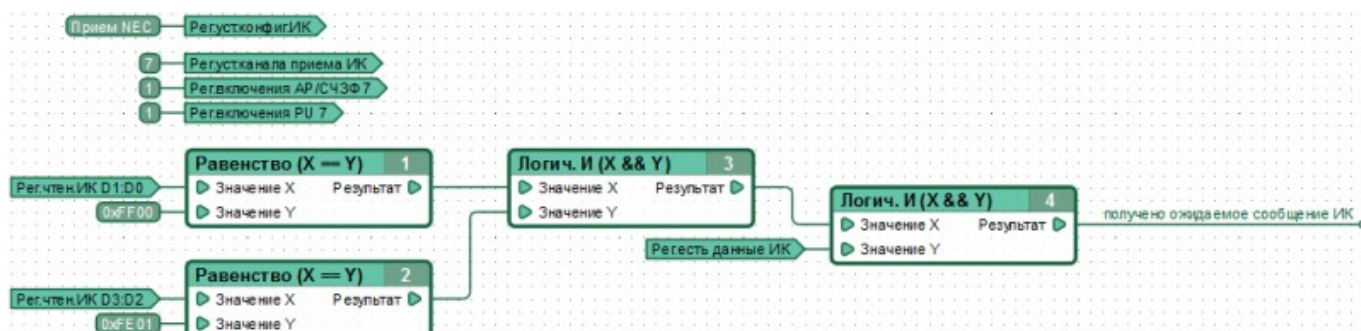
Регистр	Ожидаемые значения	
Регистр начала передачи ИК	≥ 1	= загрузить данные из регистров передачи в буфер передачи драйвера пульта ИК ДУ;
	0	= не загружать данные в буфер передачи драйвера пульта ИК ДУ.

Регистр сообщения передачи ИК D1:D0 Регистр сообщения передачи ИК D3:D2	0...0xFFFF = значения передаваемых байт сообщения драйвера пульта ИК ДУ, по два байта на регистр.
--	---

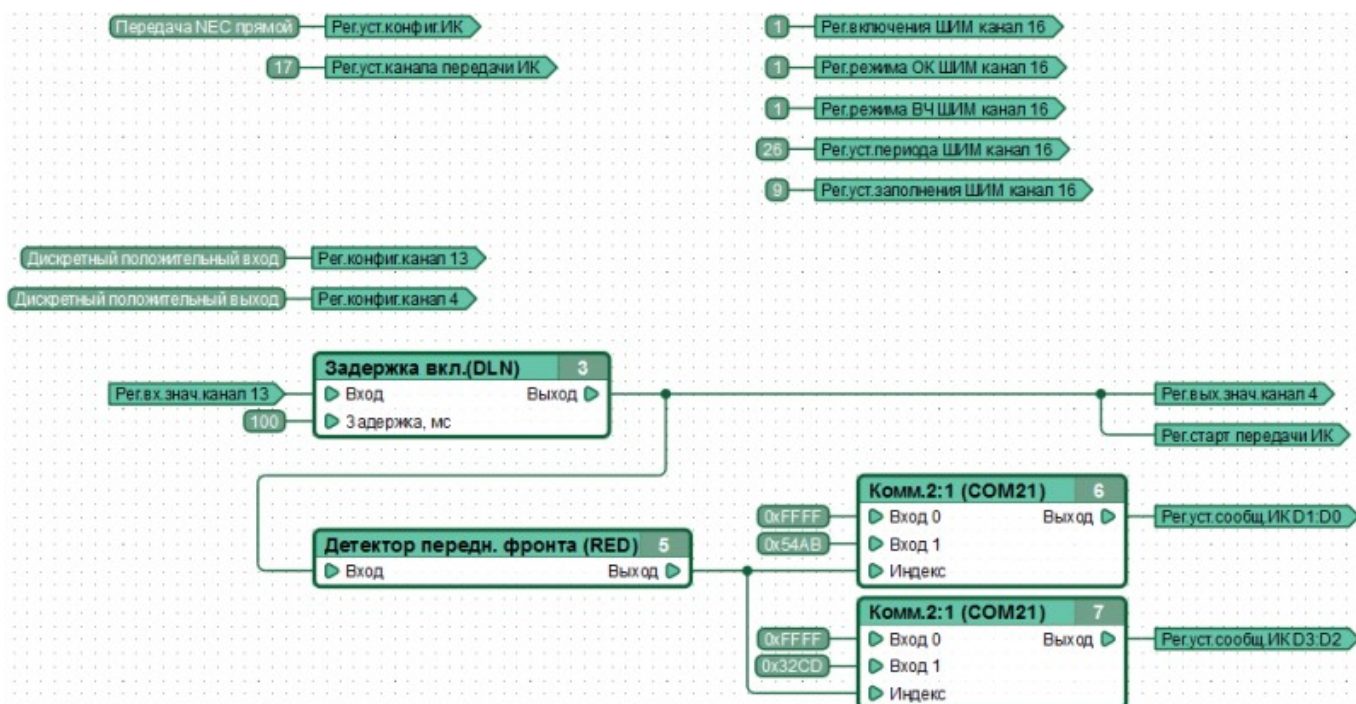
Специальная команда стандарта NEC – «повтор команды», кодируется значениями D1:D0 = 0xFFFF и D3:D2 = 0xFFFF как при приеме так и при передаче.

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.

Пример функциональной диаграммы приема контроллером команд от внешнего ИК-пульта ДУ. Контроллер ожидает получения сообщения, содержащего номер адреса 0x00 и команду с кодом 0x01. Проверка полученных от пульта ИК ДУ команд на соответствие ожидаемым выполняется путем их сравнения с константами, содержащими в младшем байте требуемое значение, а в старшем — его инверсную версию.



Пример функциональной диаграммы эмуляции контроллером CANNY 5 nano ИК-пульта ДУ, т.е. передачи ИК-команд управляемому устройству. При появлении на входе канала №13 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.



Примечание: В примере отправки ИК-сообщения также реализовано использование канала №16 контроллера, работающего в режиме ВЧ ШИМ с периодом 26мкс и заполнением 9мкс, генерирующего установленный стандартом NEC сигнал несущей частоты.

12 CANNY 5 nano, Аналого-цифровой преобразователь (АЦП)

12.1 Общее описание

Восемь из восемнадцати каналов ввода-вывода CANNY 5 nano, а именно каналы №№10, 11, 12, 13, 14, 15, 16, 17, могут быть использованы в качестве независимых друг от друга аналого-цифровых преобразователей. Кроме того, контроллер имеет отдельный АЦП встроенного термодиода.

Для активации драйвера АЦП на соответствующем канале необходимо передать значение, не равное «0», в соответствующий каналу «Регистр включения аналого-цифрового преобразователя канала №XX».

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.

Доступные пользователю каналы АЦП имеют разрешение 12 бит.

12.2 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:

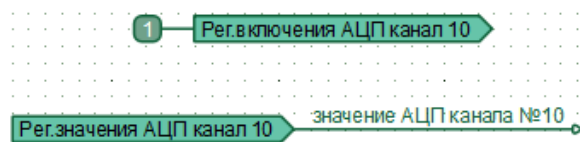
Регистр	Ожидаемые значения	
Регистр включения аналого-цифрового преобразователя канала №10	≥ 1	= подключить канал к драйверу АЦП;
...	0	= отключить канал от драйвера АЦП (канал находится под управлением драйвера ввода-вывода и доступен для использования в качестве дискретного входа/выхода).
Регистр включения аналого-цифрового		

преобразователя канала №17 Регистр включения аналого-цифрового преобразователя внутреннего термодиода	
--	--

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

Регистр	Возвращаемые значения
Регистр чтения АЦП внутреннего термодиода Регистр значения АЦП канала №10 ... Регистр значения АЦП канала №17	0...4095 = результат работы аналого-цифрового преобразователя (измерений напряжения аналогового сигнала) соответствующего канала АЦП.

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



Примечание: Аналого-цифровые преобразователи контроллера CANNY 5 папо являются линейными.

Примечание: Измерение значений напряжения каждого канала производится контроллерам один раз за 10 мс. В промежутке между измерениями регистр значения АЦП соответствующего канала сохраняет предыдущее измеренное значение.

13 CANNY 5 nano, Драйвер датчика DHT11-DHT22-AM2302

13.1 Общее описание

Контроллер CANNY 5 nano может работать с датчиками относительной влажности и температуры DHT11 / DHT22 / AM2302, используя однопроводное подключение с последовательным протоколом обмена данными, при этом контроллер выступает в качестве ведущего устройства, генерируя запросы к датчику, а DHT11 / DHT22 / AM2302 - в качестве ведомого, передавая данные замеров влажности и температуры, отвечая на данные запросы.

Для подключения контроллера CANNY 5 nano к DHT11 / DHT22 / AM2302 может использоваться любой из его каналов ввода-вывода, таким образом к контроллеру может быть одновременно подключено до 18 таких датчиков - по одному датчику на каждый канал контроллера, однако драйвер может одновременно работать только с одним датчиком, т.е. при работе с несколькими датчиками DHT11 / DHT22 / AM2302 необходимо организовывать их последовательный опрос.

В зависимости от типа и конструктивных особенностей применяемого датчика (наличия встроенной подтяжки к 5В), может потребоваться подтяжка данного канала снаружи к напряжению 5В резистором номиналом от 3 кОм до 7 кОм, либо использование внутреннего подтягивающего резистора, при его наличии у используемого канала (для каналов №№7,8,9,13,16,17).

Драйвер датчика DHT11-DHT22-AM2302 в своей работе использует ресурсы каналов контроллера, но имеет более высокий приоритет чем драйвера ввода-вывода. Таким образом, при активации драйвера, для задействованных в его работе каналов, изменение значений в связанных с ними регистрах драйвера ввода-вывода будет проигнорировано контроллером.

13.2 Регистры драйвера датчика DHT11-DHT22-AM2302

Ниже приведено описание регистров управления работой драйвера датчика DHT11-DHT22-AM2302.

Регистры конфигурации драйвера датчика DHT11-DHT22-AM2302.

Для активации драйвера необходимо установить номер канала контроллера, подключенного к датчику с которым предполагается начать работу, в соответствующий

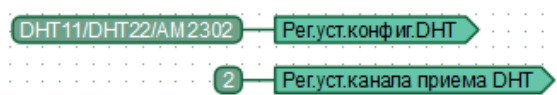
регистр драйвера.

Примечание: В каждый отдельный момент времени драйвер может работать только с одним из подключенных к нему датчиков DHT11-DHT22-AM2302, при этом допускается динамическое переключение между подключенными датчиками в процессе выполнения функциональной диаграммы.

Отправка запросов на получение данных датчиков выполняется автоматически, 1 раз в 3 секунды, и начинается сразу после активации драйвера. Контроль получения ответов DHT11-DHT22-AM2302 выполняется с помощью соответствующих регистров драйвера.

Примечание: Во избежание потери данных при работе с несколькими датчиками DHT11-DHT22-AM2302, выполняйте переключение между ними только после получения, обработки или сохранения данных от текущего активного датчика.

Пример конфигурации канала №2 для работы с датчиком DHT11-DHT22-AM2302.



Регистры конфигурации драйвера датчика DHT11-DHT22-AM2302.

Регистр	Ожидаемые значения
Регистр конфигурации DHT	1...N = активизация драйвера и установка типа подключаемого устройства (задается специальной константой из справочника констант); 0 = отключить канал от драйвера датчика DHT11-DHT22-AM2302, вернуть управление каналом драйверу каналов ввода-вывода и разрешить изменения его состояния из функциональной диаграммы.
Регистр номера канала приема датчика DHT	0...17 = установить номер канала контроллера используемого для работы с датчиком DHT11-DHT22-AM2302.

Конфигурация канала для работы в данном режиме задается следующим именованными константами:

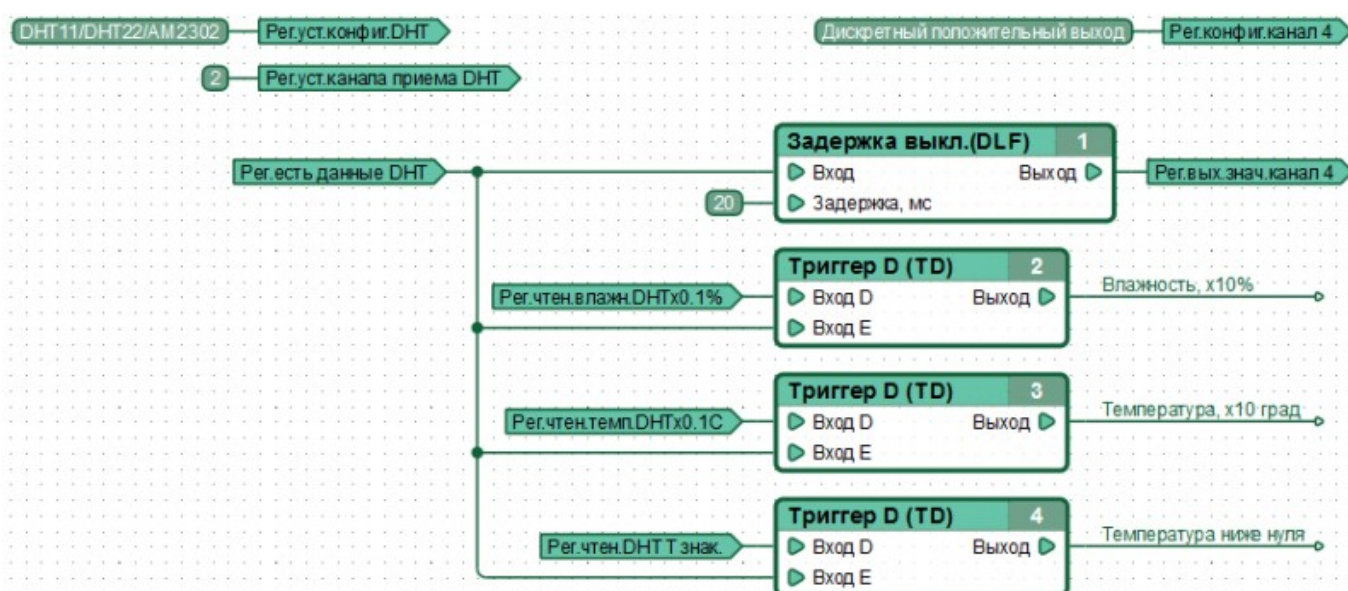
Параметр	Перечень допустимых значений
Тип устройства	Датчик температуры и относительной влажности DHT11/DHT22/AM2302

Регистры приема драйвера датчика DHT11-DHT22-AM2302.

В процессе работы, при получении от подключенного датчика ответов на запросы контроллера, данные размещаются в его соответствующие регистры приема.

Регистр	Возвращаемые значения	
Регистр наличия принятых данных DHT	1	= данные от датчика успешно получены и доступны в регистрах чтения драйвера датчика DHT11-DHT22-AM2302;
	0	= в буфере приема драйвера отсутствуют актуальные данные.
Регистр чтения показаний датчика DHT: относительная влажность, в десятых долях процента	0...999	= значения измеренной датчиком относительной влажности, в десятых долях процента.
Регистр чтения показаний датчика DHT: модуль температуры в десятых долях градусов Цельсия	0...800	= модуль значения измеренной датчиком температуры, в десятых долях градусов Цельсия.
Регистр чтения показаний датчика DHT: знак температуры (0 = выше нуля; 1 = ниже нуля)	0	= измеренная датчиком температура не ниже нуля;
	1	= измеренная датчиком температура ниже нуля.

Пример функциональной диаграммы работы с датчиком DHT22.



При получении данных от датчика DHT, подключенного к каналу №2 контроллера, в регистре наличия принятых данных драйвера DHT появляется значение «1» и данные из регистров чтения показаний температуры, с помощью D-триггеров, сохраняются в соответствующие именованные сети для дальнейшей обработки. Получение данных

подтверждается кратковременным включением контрольного светодиода контроллера (канал №4).