

FANtastic

FN-603

**МНОГОКАНАЛЬНЫЙ КОНТРОЛЛЕР
ВЕНТИЛЯТОРОВ**



руководство пользователя

1. Описание устройства

1.1. Технические характеристики

- количество каналов регулирования – 6
- номинальное напряжение питания вентиляторов – 12 В
- максимальный потребляемый одним вентилятором ток – 0.7 А
- диапазон измерения температуры – -40...+125 °С.
- дискретность измерения температуры – 0.1°С.
- диапазон измерения скорости вращения вентиляторов – 0...6000 об/мин
- начальный разгон вентиляторов
- звуковая сигнализация аварии
- тип интерфейса – USB
- совместимость с ОС – Windows 98/ME/2000/XP, Linux

1.2. Назначение устройства

Многоканальный контроллер вентиляторов FN-603 предназначен для регулировки скорости вращения вентиляторов, служащих для охлаждения узлов персонального компьютера. Контроллер устанавливается внутрь системного блока, к нему подключается до 6-ти вентиляторов и датчиков температуры. В зависимости от температуры охлаждаемого объекта контроллер устанавливает скорость вращения вентилятора согласно заданной регулировочной характеристике. Параметры регулирования задаются с помощью специальной управляющей программы и передаются в контроллер по интерфейсу USB. Это позволяет обойтись вообще без органов управления на передней панели устройства, что облегчает его монтаж в корпусе компьютера. Все параметры регулирования сохраняются в энергонезависимой памяти устройства, поэтому после их задания контроллер может работать автономно, без вмешательства компьютера. При каждом включении питания параметры автоматически загружаются из энергонезависимой памяти. Это обеспечивает работоспособность контроллера даже на этапе загрузки ОС, или под теми ОС, которые не поддерживаются ПО контроллера (например, MS-DOS).

1.3. Комплектность

- плата контроллера – 1 шт.
- датчик температуры – 6 шт.
- переходник для вентилятора – 4 шт
- переходник для вентилятора с отводом таходатчика – 2 шт

2. Установка устройства

2.1. Установка аппаратуры

Плата контроллера устанавливается в любой свободный 3.5-дюймовый отсек системного блока. К плате необходимо подключить питание +5 В и +12 В. Для этого используется любой свободный выход блока питания, имеющий стандартный разъем (аналогичный разъему питания жестких дисков). Датчики температуры устанавливаются на радиаторах или корпусах охлаждаемых узлов и подключаются к разъемам THERM1...THERM6 платы контроллера. Вентиляторы подключаются к разъемам FAN1...FAN6 платы контроллера. Если необходимо сохранить функции тахометра, реализованного на материнской плате, для подключения вентиляторов нужно использовать

специальный переходник с отводом таходатчика (два таких переходника входят в комплект устройства). При подключении платы необходимо обеспечить соответствие номера датчика температуры номеру вентилятора. Датчик, подключенный к разъему THERM1, должен быть установлен на том радиаторе, который охлаждается вентилятором FAN1 и так далее. Подключение к порту USB осуществляется внутри корпуса компьютера, для чего используются любые свободные штырьки порта USB на материнской плате.

2.2. Установка программного обеспечения

После того, как плата контроллера установлена в компьютер и к ней подключены датчики температуры и вентиляторы, нужно включить компьютер и произвести установку программного обеспечения. Процесс установки зависит от используемой операционной системы.

2.2.1. Установка программного обеспечения под Windows 98/ME/2K/XP

После загрузки Windows на экране появится сообщение о том, что найдено новое оборудование «FANtastic™ FN-603 Fan Controller». В первую очередь нужно установить драйвер, который находится в директории Soft\Driver поставляемого диска. После успешной установки драйвера нужно запустить управляющую программу Fantastic.exe.

2.3. Обновление встроенного программного обеспечения (firmware)

Для обновления встроенного программного обеспечения контроллера нужно запустить специальную утилиту FanUpgrade.exe и следовать инструкциям.

3. Техническое описание устройства

3.1. Описание логики работы контроллера

3.1.1. Измерение температуры

Измерение температуры производится с помощью интегральных датчиков, которые должны быть закреплены на тех узлах компьютера, для охлаждения которых используются вентиляторы. Датчик температуры и вентилятор, подключенные к одному каналу контроллера, должны быть установлены на одном и том же узле компьютера. Важно обеспечить низкое тепловое сопротивление между нагревающимся узлом и датчиком температуры.

3.1.2. Управление скоростью вращения вентиляторов

Для управления скоростью вращения вентиляторов производится регулировка напряжения питания с помощью ключевых стабилизаторов. Для того чтобы процесс регулирования скорости вращения не сопровождался увеличением акустического шума, для питания вентиляторов используется не ШИМ-сигнал, а его постоянная составляющая после фильтрации. При изменении управляющего кода от 0 до 255 напряжение питания вентилятора меняется от 0 до 12 В, т.е. от 0 до 100% относительно номинального значения. При этом скорость вращения вентилятора также меняется от 0 до 100% относительно номинального значения. Зависимость, связывающая скорость вращения и напряжение питания, в общем случае не является линейной. Однако приблизительно его можно считать линейным, если отбросить начальный участок характеристики вблизи нуля. Для задания регулировочной характеристики используются 4 параметра: минимальная температура TempMin, минимальная скорость PWMMin, максимальная температура TempMax и максимальная скорость PWMMax. Вид задаваемой регулировочной кривой показан на

рисунке 1.

Начальный участок кривой является горизонтальным. Это означает, что при температуре ниже TempMin вентилятор вращается на постоянной скорости PWMMin. Для этого участка можно выбрать минимальную скорость вентилятора, на которой он надежно вращается.

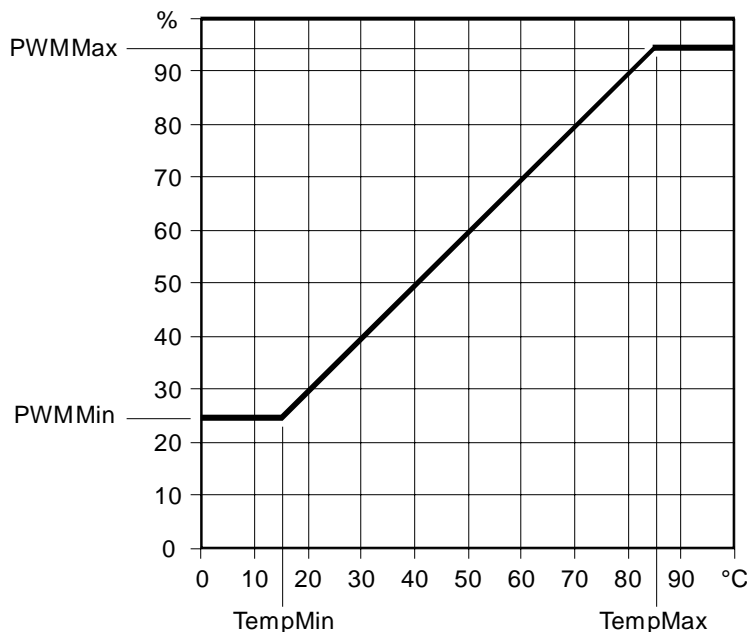


Рисунок 1. Вид регулировочной кривой.

Средний участок кривой имеет постоянный наклон. Это означает, что когда температура повышается от TempMin до TempMax, скорость вентилятора увеличивается от PWMMin до PWMMax.

Последний участок кривой тоже горизонтальный. Это означает, что при температуре выше TempMax вентилятор вращается на постоянной скорости PWMMax. Для этого участка целесообразно выбрать максимальную скорость вентилятора.

3.1.3. Принудительный разгон вентиляторов

Для каждого из каналов отдельно может быть включен режим принудительного разгона вентиляторов при включении питания. В этом режиме сразу после включения питания контроллера в течение 2.5 секунд на вентилятор подается полное напряжение питания. Необходимость такого режима связана с тем, что при очень низких установленных начальных скоростях вентилятор может не стартовать. Если же его принудительно разогнать, после этого он нормально работает на низких оборотах.

3.1.4. Аварийная сигнализация

Контроллер вентиляторов имеет возможность автономной аварийной сигнализации. Для каждого канала можно включить сигнализацию остановки вентилятора и сигнализацию перегрева.

Для сигнализации остановки вентилятора индивидуально для каждого канала задается минимально допустимая скорость вращения. Если измеренная скорость вращения окажется ниже, включается звуковая сигнализация аварии в виде коротких звуковых сигналов.

Для сигнализации перегрева для каждого канала индивидуально задается

температурный порог. Если этот порог будет превышен, то включается звуковая сигнализация аварии в виде коротких звуковых сигналов.

3.2. Описание аппаратной части контроллера

Основой контроллера вентиляторов является микроконтроллер ATmega88 фирмы «Atmel». Микроконтроллер с помощью встроенных модулей PWM управляет регулируемые ключевыми стабилизаторами, которые формируют напряжение питания вентиляторов.

Для измерения скорости вращения используется сигнал таходатчика, поступающий от вентилятора. Для тех моделей вентиляторов, которые не имеют такого сигнала, скорость вращения измеряться не будет.

Для измерения температуры могут быть использованы либо аналоговые интегральные датчики температуры TMP36 фирмы «Analog Devices», либо цифровые датчики температуры DS18S20 фирмы «Maxim». Тип датчиков зависит от варианта исполнения контроллера. Датчики подключаются к контроллеру с помощью 3-х проводного шлейфа и могут быть расположены в любом месте системного блока компьютера, и даже за его пределами.

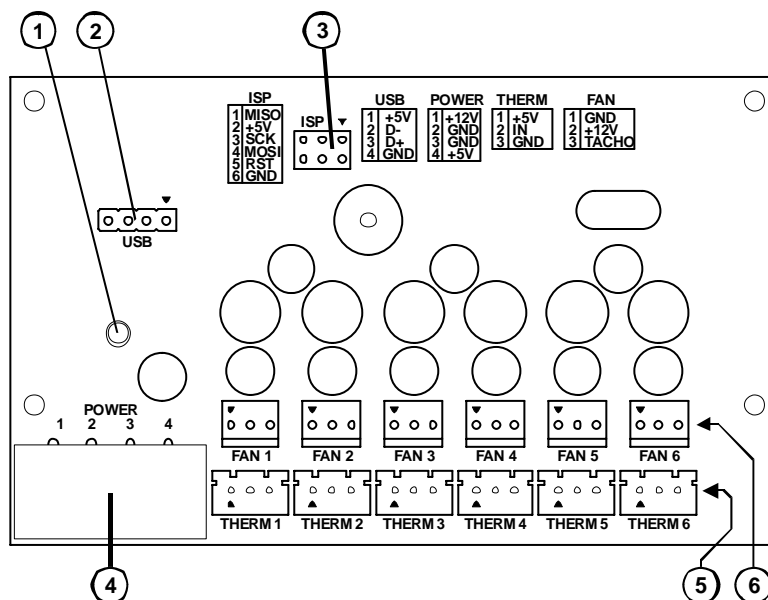


Рисунок 2. Внешний вид платы контроллера:

1 – светодиод индикации питания; 2 – разъем для подключения интерфейса USB; 3 – разъем для начального программирования микроконтроллера; 4 – разъем питания; 5 – разъемы для подключения датчиков температуры; 6 – разъемы для подключения вентиляторов.

Порт USB реализован с помощью микросхемы FT232R фирмы «FTDI». Для подключения к порту USB компьютера на плате контроллера установлены штырьки, которые с помощью 4-х проводного шлейфа необходимо соединить со штырьками порта USB на материнской плате компьютера. Кроме того, на плате контроллера предусмотрено место для установки стандартного разъема типа USB-B, который можно использовать для подключения к порту USB стандартным кабелем.

Для питания контроллера используются напряжения +5 В и +12 В со штатного блока питания компьютера. Стандартный разъем для подключения питания установлен на плате контроллера. Внешний вид платы контроллера показан на рис. 2.

3.3. Описание управляющей программы

Контроллер выполняет процесс регулирования автономно и не требует постоянной работы управляющей программы. Программа необходима лишь для загрузки параметров регулирования и для контроля текущих значений параметров. Главное окно управляющей программы имеет пять закладок: «Common», «Channel», «History», «Presets» и «Settings». Закладка «Common» (рис. 3) служит для отображения текущей температуры, скорости вращения вентиляторов и напряжения питания вентиляторов для всех каналов одновременно.

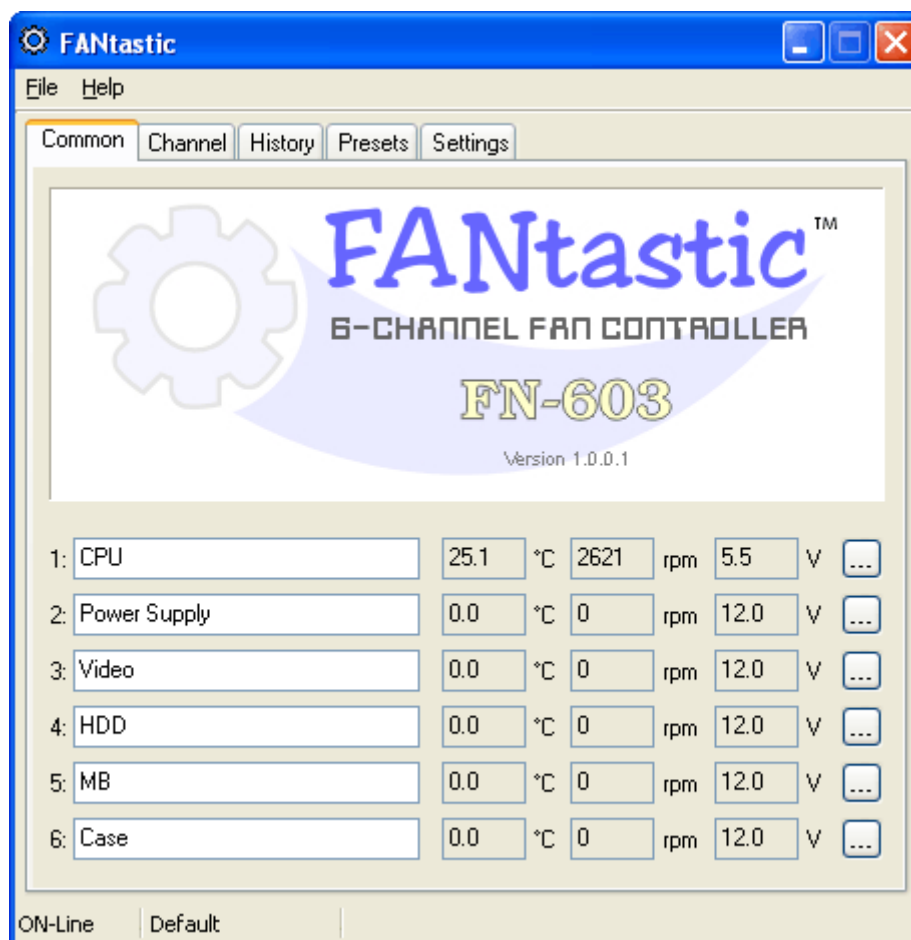


Рисунок 3. Закладка «Common» управляющей программы.

Для каждого канала имеется поле ввода, где можно задать название канала, например, «CPU», «Video», «HDD», «Power Supply», «Case» и т.д. Это название сохраняется в ini-файле и отображается при каждом запуске программы.

Если для любого из каналов включен контроль аварийной температуры или аварийной скорости вращения, то при достижении аварийных условий поле вывода данного параметра меняет цвет на красный.

Кнопки, расположенные справа от полей вывода, позволяют сразу перейти к настройкам выбранного канала.

Закладка «Channel» (рис. 4) служит для ввода характеристик регулирования канала. Номер канала выбирается из выпадающего списка **Channel**. Над графиком отображается текущая температура, скорость вращения вентилятора, напряжение питания вентилятора и значение PWM для данного канала.

Группа **Auto** позволяет задать параметры автоматического регулирования. В полях **T_min**, **PWM_min**, **T_max** и **PWM_max** задаются минимальные и максимальные значения температуры и PWM для кривой автоматического регулирования. Если установлен флажок **Enable Auto Control**, то производится автоматическое регулирование. Если установлен флажок **Enable startup boost**, то производится принудительный разгон вентиляторов при включении питания.

Если автоматическое регулирование выключено, то скорость вентилятора устанавливается вручную. В этом режиме значение PWM в процентах можно вводить вручную. Можно также мышью перемещать на графике зеленую горизонтальную линию, при этом будет меняться значение PWM.

Группа **Alarm** служит для управления аварийной сигнализацией. В поле **T_alarm** можно задать значение аварийной температуры, и если установлен флажок **Enable T alarm**, то при достижении этой температуры сработает аварийная звуковая сигнализация. В поле **V_alarm** можно задать значение аварийной скорости вращения вентилятора, и если установлен флажок **Enable V alarm**, то при снижении скорости вращения ниже этого порога сработает аварийная звуковая сигнализация.

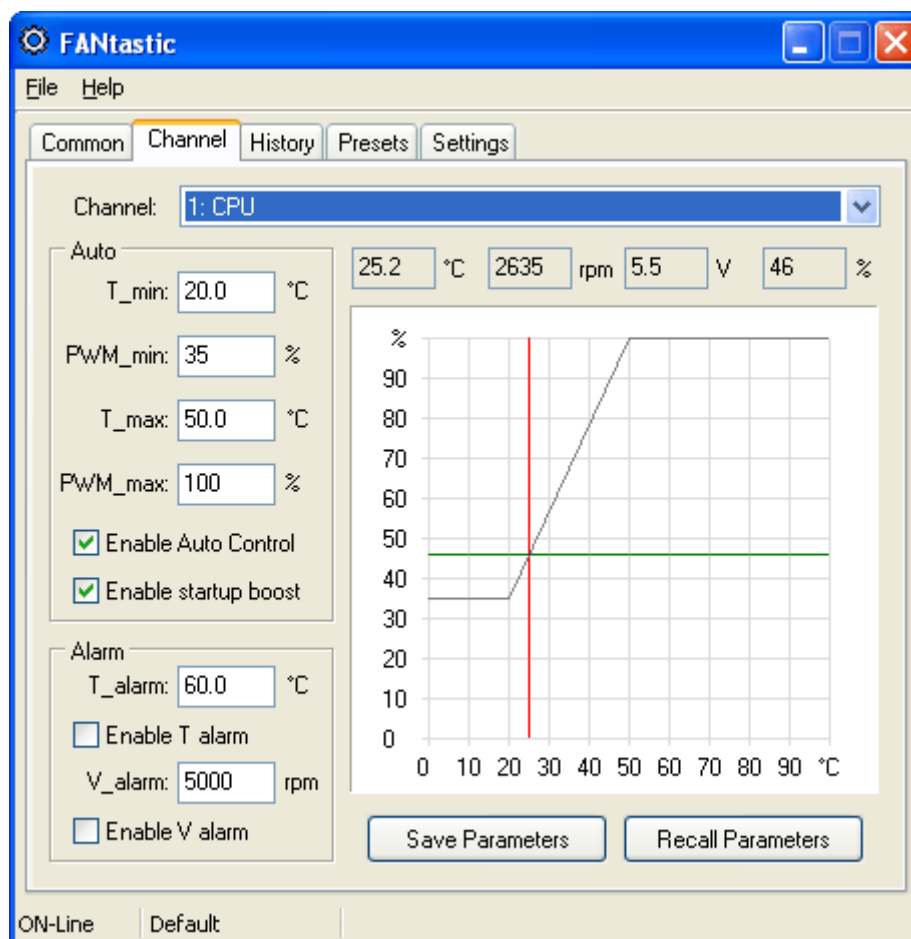


Рисунок 4. Закладка «Channel» управляющей программы.

Для того чтобы сохранить в энергонезависимой памяти параметры, заданные на закладке **Channel**, нужно нажать кнопку **Save Parameters**. После этого сохраненные параметры будут автоматически загружаться при включении питания контроллера. Чтобы убедиться в правильности установленных параметров, нужно нажать кнопку **Recall Parameters**. При этом значения параметров считываются из контроллера и отображаются в

соответствующих полях.

Закладка «History» (рис. 5) позволяет в графическом виде посмотреть историю изменения температуры для каждого из каналов. Цвет каждого из графиков можно изменить, для чего необходимо сделать двойной щелчок на нужной строке таблицы.

Дополнительно в таблице отображается текущая, средняя, минимальная и максимальная температура для каждого канала.

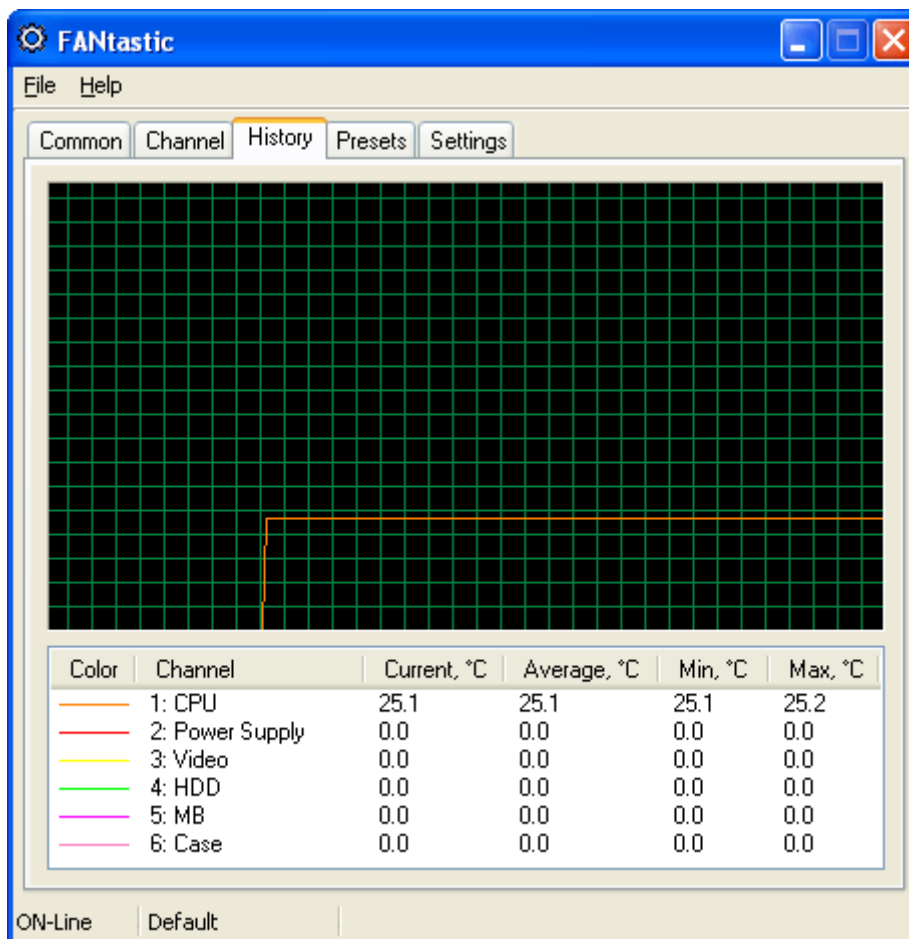


Рисунок 5. Закладка «History» управляющей программы.

Закладка «Presets» (рис. 6) позволяет выбрать одну из существующих предустановок настроек контроллера, а также создать или удалить предустановку.

Предустановки – это наборы настроек контроллера, наиболее подходящие для разных режимов работы компьютера. Например, можно создать предустановку «Игровая», когда будет осуществляться интенсивное охлаждение, или предустановку «Ночная», которая при небольшой загрузке процессора позволит снизить шум.

Для создания предустановки нужно нажать кнопку **Add**, затем ввести имя предустановки. При сохранении предустановки будет сохранен текущий набор настроек контроллера. Для создания другой предустановки нужно поменять режимы работы контроллера и снова проделать описанные выше действия.

Для того чтобы выбранная предустановка вступила в силу, нужно нажать кнопку **Active**. При этом предустановка становится активной, ее имя в списке становится жирным. Имя активной предустановки также выводится в строке состояния.

Имя любой предустановки можно изменить, нажав кнопку **Edit**.

Для того чтобы удалить предустановку, нужно нажать кнопку **Delete**. Предустановку по умолчанию (default) удалить нельзя.

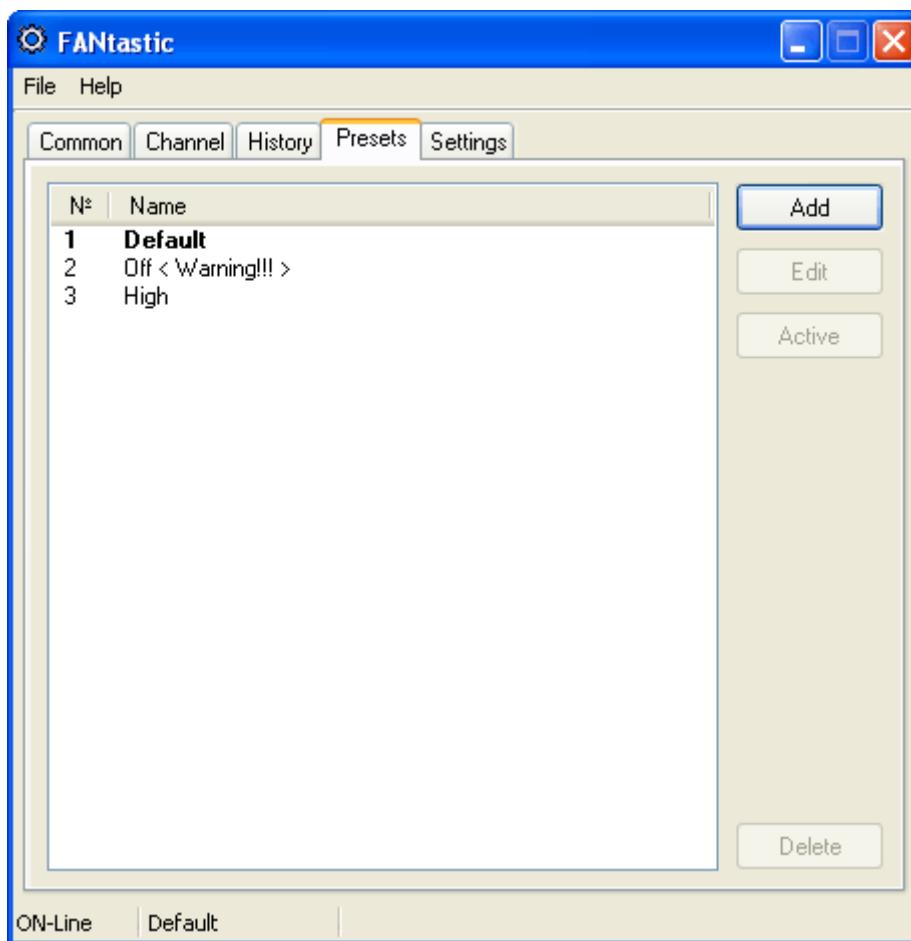


Рисунок 6. Закладка «Presets» управляющей программы.

Закладка «Settings» (рис. 7) служит для настройки управляющей программы контроллера.

Группа **Hints** служит для выбора режима отображения подсказок.

Если установлен флажок **Show tool tips**, то будут выводиться подсказки при наведении курсора мыши на любой элемент управления.

Если установлен флажок **Show hints in status bar**, то при наведении курсора мыши на любой элемент управления соответствующая подсказка будет выводиться в строке состояния.

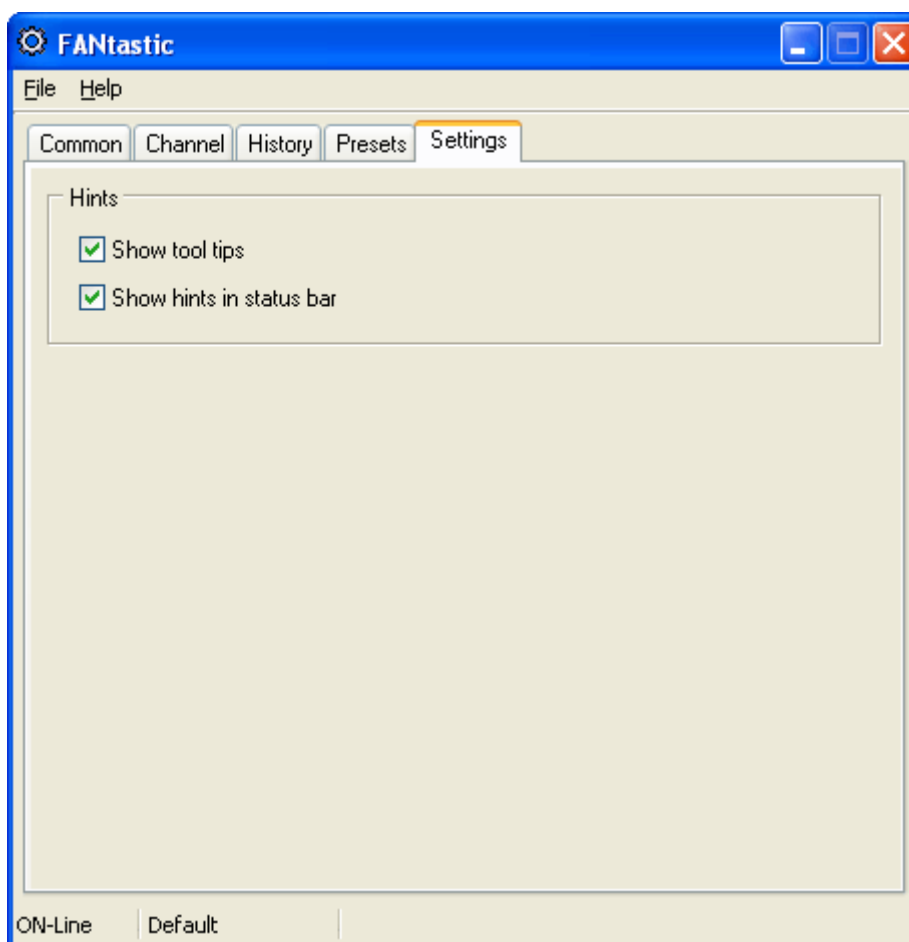


Рисунок 7. Закладка «Settings» управляющей программы.

В строке состояния внизу главного окна программы дополнительно отображается состояние связи с контроллером: «On-Line», если связь установлена, и «Off-Line», если связи нет.

3.4. Описание динамической библиотеки fn603.dll

Динамическая библиотека fn603.dll реализует команды протокола в виде отдельных функций. Кроме того, библиотека содержит дополнительные функции, предназначенные для настройки порта. Список функций библиотеки fn603.dll приведен ниже:

3.4.1. Функция FN603_OpenDevice

Открывает USB-устройство с именем “FN-603”.

bool FN603_OpenDevice(void)

В случае успешного выполнения возвращает **true**.

3.4.2. Функция FN603_CloseDevice

Закрывает открытое ранее USB-устройство.

bool FN603_CloseDevice(void)

В случае успешного выполнения возвращает **true**.

3.4.3. Функция FN603_GetLastError

Чтение строки с информацией о последней ошибке обмена с устройством.

void FN603_GetLastError(LPCSTR &lpcStr)

Возвращает указатель на строку, которая содержит информацию о последней ошибке. Если предыдущая операция обмена с устройством прошла без ошибок, возвращается указатель на пустую строку.

3.4.4. Функция FN603_GetInfo

Чтение информации об устройстве.

bool FN603_GetInfo(LPCSTR &lpcStr)

Возвращает указатель на строку, которая содержит информацию об устройстве. В данном случае строка выглядит следующим образом: “FN-603 V1.0”, где “FN-603” – тип контроллера, “V1.0” – версия firmware 1.0.

В случае успешного выполнения возвращает **true**.

3.4.5. Функция FN603_GetT

Чтение текущей температуры для всех каналов.

bool FN603_GetT(int &t1, int &t2, int &t3, int &t4, int &t5, int &t6)

Возвращает значения температуры **t1**, **t2**, **t3**, **t4**, **t5**, **t6** для всех 6-ти каналов в виде целого знакового числа. Значения могут лежать в диапазоне –400...+1250, для перевода в единицы температуры их необходимо умножить на 0.1°C. Если в каком-то из каналов термометр не подключен, показания всегда будут равны –40.0°C.

В случае успешного выполнения возвращает **true**.

3.4.6. Функция FN603_GetV

Чтение текущей скорости вращения вентиляторов для всех каналов.

bool FN603_GetV(int &v1, int &v2, int &v3, int &v4, int &v5, int &v6)

Возвращает значения скорости вращения вентиляторов **v1**, **v2**, **v3**, **v4**, **v5**, **v6** для всех 6-ти каналов в виде целого беззнакового числа. Значения могут лежать в диапазоне 0...9999 и представлены в оборотах в минуту. Если в каком-то из каналов вентилятор не подключен или не имеет таходатчика, показания всегда равны 0.

В случае успешного выполнения возвращает **true**.

3.4.7. Функция FN603_SetM

Установка режима работы одного канала.

bool FN603_SetM(int chan, int mode, int Tmin, int Pmin, int Tmax, int Pmax, int At, int Av, int Pman)

Функция имеет следующие параметры:

chan – номер канала. Может принимать значения 0...5.

mode – режим работы канала, содержит набор управляющих битов.

Бит **mode.0** – разрешение автоматического регулирования. Если равен единице, производится управление скоростью вращения вентилятора согласно заданной регулировочной кривой. Если равен нулю, скорость вращения вентилятора постоянна и равна заданной параметром **Pman**.

Бит **mode.1** – разрешение начального разгона вентилятора. Если равен единице, при включении питания производится разгон вентилятора до максимальной скорости, а затем переход на требуемую скорость. Если равен нулю, начальный разгон не производится, и сразу устанавливается требуемая скорость.

Бит **mode.2** – разрешение тревоги по превышению температуры. Если равен единице, в случае превышения температурой заданного порога включается звуковая сигнализация аварии и вентилятор переходит на полные обороты. Если равен нулю, контроль температурного порога не производится.

Бит **mode.3** – разрешение тревоги по уменьшению скорости вращения вентилятора. Если равен единице, в случае уменьшению скорости вращения вентилятора ниже установленного порога включается звуковая сигнализация аварии и вентилятор переходит на полные обороты. Если равен нулю, контроль скорости вращения не производится.

Tmin – температура, выше которой начинается разгон вентилятора. Может принимать значения -400...+1250, что соответствует температуре -40.0...+125.0°C.

Pmin – величина PWM, с которой начинается разгон вентилятора. Может принимать значения 0...255, что соответствует напряжению питания вентилятора 0...100%.

Tmax – температура, до которой продолжается разгон вентилятора. Может принимать значения -400...+1250, что соответствует температуре -40.0...+125.0°C.

Pmax – величина PWM, до которой продолжается разгон вентилятора. Может принимать значения 0...255, что соответствует напряжению питания вентилятора 0...100%.

At – аварийная температура. Может принимать значения -400...+1250, что соответствует температуре -40.0...+125.0°C. Аварийная температура контролируется, если бит **mode.2** равен единице.

Av – аварийная скорость вращения вентиляторов. Может принимать значения 0...9999 оборотов в минуту. Аварийная скорость вращения контролируется, если бит **mode.3** равен единице.

Pman – величина PWM ручного режима работы вентилятора.

В случае успешного выполнения возвращает **true**.

3.4.8. Функция FN603_GetM

Чтение текущего режима работы одного канала.

bool FN603_GetM(int chan, int &mode, int &Tmin, int &Pmin, int &Tmax, int &Pmax, int &At, int &Av, int &Pman)

Функция имеет следующие параметры:

chan – номер канала. Может принимать значения 0...5.

Возвращаемые значения соответствуют параметрам функции **FN603_SetM**.

В случае успешного выполнения возвращает **true**.

3.4.9. Функция FN603_SetPWM

Установка величины PWM (напряжения питания) вентиляторов для всех каналов.

bool FN603_SetPWM(int p1, int p2, int p3, int p4, int p5, int p6)

Функция имеет следующие параметры:

p1, p2, p3, p4, p5, p6 – величина PWM для каналов 1...6. Может принимать значения 0...255, что соответствует напряжению питания вентилятора 0...100%.

В случае успешного выполнения возвращает **true**.

3.4.10. Функция FN603_GetPWM

Чтение текущих значений величины PWM для всех каналов.

bool FN603_GetPWM(int &p1, int &p2, int &p3, int &p4, int &p5, int &p6)

Возвращаемые значения соответствуют параметрам функции **FN603_SetPWM**.

В случае успешного выполнения возвращает **true**.

3.4.11. Функция FN603_SavePar

Сохранение всех параметров выбранного канала в EEPROM.

bool FN603_SavePar(int chan)

Функция имеет следующие параметры:

chan – номер канала. Может принимать значения 0...5.

В случае успешного выполнения возвращает **true**.

3.4.12. Функция FN603_CallPar

Вызов всех параметров выбранного канала из EEPROM.

bool FN603_CallPar(int chan)

Функция имеет следующие параметры:

chan – номер канала. Может принимать значения 0...5.

В случае успешного выполнения возвращает **true**.

3.5. Описание протокола обмена между контроллером и компьютером

Многоканальный контроллер вентиляторов FN-603 подключается к компьютеру с помощью интерфейса USB. Команды передаются в виде пакетов согласно протоколу WAKE, скорость обмена 38400 бод. Инициатором обмена всегда выступает компьютер. В ответ на каждую команду контроллер передает пакет, который содержит тот же номер команды, а в качестве первого байта данных передается код ошибки (за исключением команд CMD_ECHO и CMD_INFO). Код ошибки 00h означает успешное выполнение команды. Любой отличный код – наличие ошибки (см. описание кодов ошибок ниже). В поле данных каждой команды передаются параметры. Для разных команд число параметров может быть разным, есть команды, которые не имеют параметров вообще.

3.5.1. Команда Cmd_Nop

Команда CMD_NOP не выполняет никакой операции. Она используется для внутренних целей и никогда не передается в контроллер или компьютер.

TX										RX											
CMD	N	D7	D6	D5	D4	D3	D2	D1	D0	CMD	N	D7	D6	D5	D4	D3	D2	D1	D0		
00h	0	-									00h	0	-								

3.5.2. Команда Cmd_Err

Контроллер передает команду CMD_ERR в качестве ответа на любую команду, если произошла ошибка приема пакета. Параметр Error Code для этой команды всегда равен ERR_TX.

TX										RX											
CMD	N	D7	D6	D5	D4	D3	D2	D1	D0	CMD	N	D7	D6	D5	D4	D3	D2	D1	D0		
01h	0	-									01h	1	Error Code								

3.5.3. Команда Cmd_Echo

Команда CMD_ECHO используется для запроса возврата пакета. Пакет может содержать до 64 байт произвольных данных. В ответ на эту команду контроллер передает пакет в неизменном виде обратно. Команда используется для проверки связи с контроллером.

TX										RX											
CMD	N	D7	D6	D5	D4	D3	D2	D1	D0	CMD	N	D7	D6	D5	D4	D3	D2	D1	D0		
02h	X	Byte1									02h	X	Byte1								
									
		ByteN											ByteN								

3.5.4. Команда Cmd_Info

Команда CMD_INFO представляет собой запрос информации о типе контроллера и версии встроенного программного обеспечения (firmware). В ответ передается пакет, содержащий 12 байт данных, которые представляют собой строку в коде ASCII: “FN-603 V1.0”, где “ FN-603” – тип контроллера, “V1.0” – версия firmware 1.0. В качестве разделителей используются пробелы (код 20h). Строка заканчивается байтом 00h.

TX										RX											
CMD	N	D7	D6	D5	D4	D3	D2	D1	D0	CMD	N	D7	D6	D5	D4	D3	D2	D1	D0		
03h	0	-									03h	12	String: “FN-603 V1.0”, 00h								

3.5.5. Команда Cmd_GETT

Команда CMD_GETT служит для считывания показаний всех датчиков температуры. Команда не имеет параметров.

Команда возвращает 6 значений температуры Temp1...Temp6. Каждое значение представлено в виде 2-х байтового целого числа со знаком и представляет собой значение температуры в градусах Цельсия, умноженное на 10. Таким образом, дискретность представления температуры составляет 0.1°C. Температура может принимать значение от –40.0 до +125.0°C, чему соответствуют коды от –400 до +1250. Если в каком-то из каналов термометр не подключен, показания всегда будут –40.0°C.

TX										RX										
CMD	N	D7	D6	D5	D4	D3	D2	D1	D0	CMD	N	D7	D6	D5	D4	D3	D2	D1	D0	
06h	0										06h	13	Error Code							
										Temp1L										
										Temp1H										
										Temp2L										
										Temp2H										
										Temp3L										
										Temp3H										
										Temp4L										
										Temp4H										
										Temp5L										
										Temp5H										
										Temp6L										
										Temp6H										

3.5.6. Команда Cmd_GETV

Команда CMD_GETV служит для считывания текущей скорости вращения всех вентиляторов. Команда не имеет параметров.

Команда возвращает 6 значений скорости вращения RPM1...RPM6. Каждое значение представлено в виде 2-х байтового целого числа без знака и представляет собой скорость вращения в оборотах в секунду. Скорость может принимать значения от 0 до 9999 оборотов в минуту. Если вентилятор какого-то из каналов не подключен или не имеет таходатчика, то для этого канала будет считываться скорость 0.

TX										RX									
CMD	N	D7	D6	D5	D4	D3	D2	D1	D0	CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
07h	0									07h	13	Error Code							
												RPM1L							
												RPM1H							
												RPM2L							
												RPM2H							
												RPM3L							
												RPM3H							
												RPM4L							
												RPM4H							
												RPM5L							
												RPM5H							
												RPM6L							
												RPM6H							

3.5.7. Команда Cmd_SETM

Команда CMD_SETM служит для установки режима работы одного канала устройства.

Параметр Chan определяет номер канала, для которого передается режим работы. Параметр Chan может принимать значения от 0 до 5, что соответствует каналам 1...6.

Первый параметр определяет режим работы канала. Этот параметр содержит битовые поля:

- Если бит A = 1, то в данном канале разрешается автоматическое регулирование скорости вентилятора согласно заданной регулировочной кривой (режим «Auto»). Если A = 0, то вентилятор данного канала работает в режиме ручного управления, скорость в этом режиме задается параметром PWMMan.
- Если бит B = 1, то в данном канале разрешается принудительный разгон вентилятора при включении устройства (режим «Boost»). В этом режиме при включении устройства на время 2 секунды вентилятор включается на полные обороты для быстрого разгона. Если B = 0, то разгон вентилятора не производится.
- Если бит TA = 1, то в данном канале разрешается контроль аварийного порога температуры. Когда измеренная температура превысит заданный порог AlarmTemp, сработает звуковая сигнализация и вентилятор перейдет на максимальную скорость. Если TA = 0, температурная сигнализация отключена.
- Если бит RA = 1, то в данном канале разрешается контроль вращения вентилятора. Когда скорость вентилятора станет ниже заданного порога AlarmRPM, сработает звуковая сигнализация и вентилятор перейдет на максимальную скорость. Если RA = 0, контроль вращения отключен.

Параметр TempMin задает минимальную температуру, с которой начинается разгон вентилятора. Параметр TempMin может принимать значения от -400 до +1250, что

соответствует температуре от -40.0 до $+125.0^{\circ}\text{C}$.

Параметр PWMMin задает минимальную скорость вентилятора, которая устанавливается при температуре менее заданной параметром TempMin. Параметр PWMMin может принимать значения от 0 до 255, что соответствует скорости вентилятора от 0 до 100% относительно номинальной.

Параметр TempMax задает максимальную температуру, после которой разгон вентилятора не производится. Параметр TempMax может принимать значения от -400 до $+1250$, что соответствует температуре от -40.0 до $+125.0^{\circ}\text{C}$.

Параметр PWMMax задает максимальную скорость вентилятора, которая устанавливается при температуре более заданной параметром TempMax. Параметр PWMMax может принимать значения от 0 до 255, что соответствует скорости вентилятора от 0 до 100% относительно номинальной. Типично значение этого параметра задается равным 255, чтобы при высоких температурах вентилятор работал на 100% скорости.

Параметр AlarmTemp задает температурный порог, превышение которого вызывает срабатывание аварийной сигнализации. Сигнализация будет срабатывать только в том случае, если она разрешена с помощью бита TA параметра Mode. Параметр AlarmTemp может принимать значения от -400 до $+1250$, что соответствует температуре от -40.0 до $+125.0^{\circ}\text{C}$.

Параметр AlarmRPM задает минимальную скорость вращения вентилятора, ниже которой произойдет срабатывание аварийной сигнализации. Сигнализация будет срабатывать только в том случае, если она разрешена с помощью бита RA параметра Mode. Параметр AlarmRPM может принимать значения от 0 до 9999 оборотов в минуту.

Параметр PWMMan задает скорость вентилятора для ручного режима работы. Параметр PWMMan может принимать значения от 0 до 255, что соответствует скорости вентилятора от 0 до 100% относительно номинальной.

		TX							
CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
08h	13	Chan							
		-	-	-	-	RA	TA	A	B
		TempMinL							
		TempMinH							
		PWMMin							
		TempMaxL							
		TempMaxH							
		PWMMax							
		AlarmTempL							
		AlarmTempH							
		AlarmRPML							
		AlarmRPML							
		PWMMan							

		RX							
CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
08h	1	Error Code							

3.5.8. Команда Cmd_GETM

Команда CMD_GETM служит для считывания текущего режима работы одного канала устройства.

Параметр Chan определяет номер канала, для которого нужно считать режим работы. Параметр Chan может принимать значения от 0 до 5, что соответствует каналам 1...6.

Команда возвращает значения, которые полностью аналогичны параметрам команды CMD_SETM (см. раздел 3.5.7).

		TX							
CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
09h	1	Chan							

		RX							
CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
09h	13	Error Code							
		-	-	-	-	RA	TA	A	B
		TempMinL							
		TempMinH							
		PWMMin							
		TempMaxL							
		TempMaxH							
		PWMMax							
		AlarmTempL							
		AlarmTempH							
		AlarmRPML							
		AlarmRPMH							
		PWMMan							

3.5.9. Команда Cmd_SETPWM

Команда CMD_SETPWM служит для задания напряжения питания вентиляторов в ручном режиме.

Параметры PWM1...PWM6 задают напряжение питания вентилятора для каждого канала устройства. Каждый из параметров может принимать значения от 0 до 255, что соответствует напряжению питания вентилятора от 0 до 100% относительно номинального. Для тех каналов, которые находятся в режиме автоматического регулирования, переданное значение игнорируется.

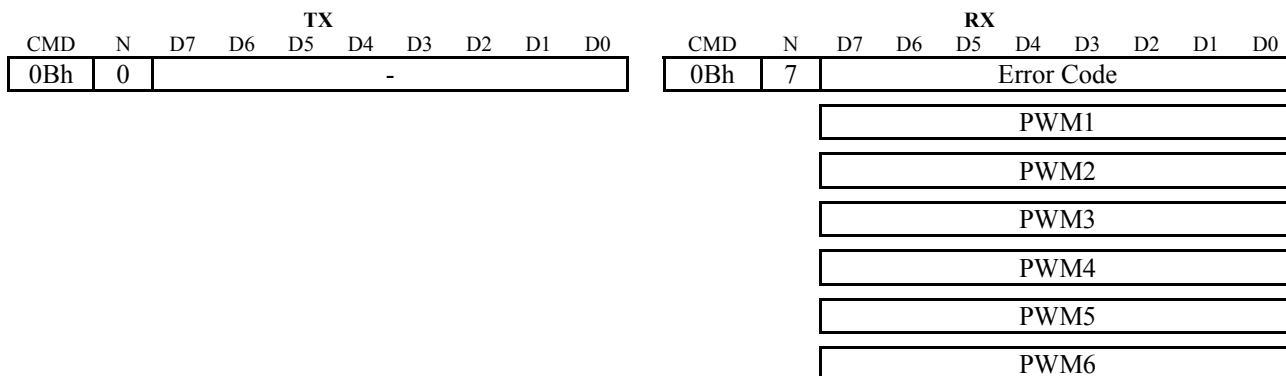
		TX							
CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
0Ah	6	PWM1							
		PWM2							
		PWM3							
		PWM4							
		PWM5							
		PWM6							

		RX							
CMD	N	D7	D6	D5	D4	D3	D2	D1	D0
0Ah	1	Error Code							

3.5.10. Команда Cmd_GETPWM

Команда CMD_GETPWM служит для считывания текущего значения напряжения питания вентиляторов. Команда не имеет параметров.

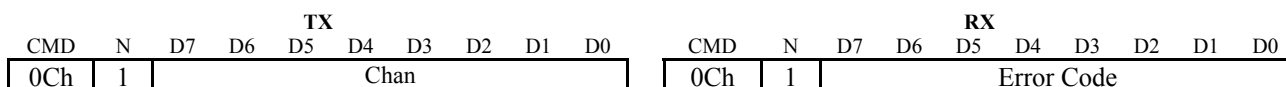
Команда возвращает значения PWM1...PWM6, которые полностью аналогичны параметрам команда CMD_SETPWM (см. раздел 3.5.9).



3.5.11. Команда Cmd_SAVEPAR

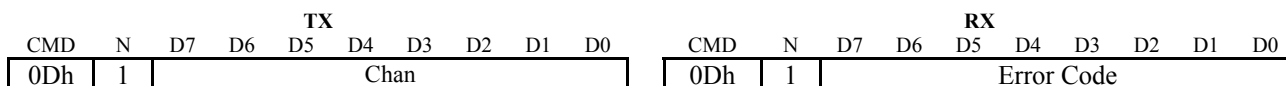
Команда CMD_SAVEPAR служит для сохранения всех параметров выбранного канала в EEPROM.

Параметр Chan определяет номер канала и может принимать значения от 0 до 5, что соответствует каналам 1...6.



3.5.12. Команда Cmd_CALLPAR

Команда CMD_CALLPAR служит для вызова всех параметров выбранного канала из EEPROM. Параметры считываются, и контроллер начинает работу согласно их значениям.



3.5.13. Коды ошибок

В случае ошибки любая команда возвращает всего один байт – код ошибки Error Code. Коды стандартных ошибок, определенных для протокола WAKE, приведены в таблице 3.5.11.

Таблица 3.5.11. Коды ошибок.

Имя ошибки	Код ошибки	Название ошибки
Err_No	00h	Нормальное завершение команды
Err_Tx	01h	Ошибка обмена с устройством
Err_Bu	02h	Устройство занято
Err_Re	03h	Устройство не готово
Err_Pa	04h	Ошибка значений параметров
Err_Nr	05h	Нет ответа
Err_Nc	06h	Нет несущей

4. Информация о производителе