Интеграция KNX-Modbus на примере ПВУ Ventmachine Orange 350.

Коробко В.Н., Науменко Н.А.

Введение.

С задачей интеграции KNX и Modbus интеграторы часто сталкиваются в проектах автоматизации зданий и умного дома. Эти два протокола исторически появились и развивались в разных сферах: KNX — в области автоматизации недвижимости, Modbus — в промышленности и энергетике. Но в инсталляционной практике нередко возникает необходимость их объединения в одной системе, например, для передачи температурных уставок из панели KNX в отопительный котёл с Modbus-управлением, или с целью управления приточной вентустановкой с Modbus-контроллером с помощью KNX интерфейсов или сенсоров.

Чтобы эффективно решать подобные задачи, интегратор должен понимать логику взаимодействия двух протоколов при обмене данными. Знание особенностей этих протоколов и методов их сопряжения часто становится практической необходимостью на многих реальных объектах.

Данная статья описывает ключевые принципы KNX-Modbus интеграции на примере подключения к шине KNX ПВУ Ventmachine Orange 350 с контроллером GTC (компания «Вентмашина», Россия) с помощью шлюза IC-M-S.1 KNX-Modbus (компания «Модуль Электроник», Россия).

Статья подготовлена с расчётом на читателя, уже знакомого с технологией KNX, как минимум на базовом уровне. Поэтому вводная часть статьи целиком посвящена основам протокола Modbus RTU, поскольку эта тема часто остаётся незнакомой для KNX-специалистов.

Также выражаем нашу искреннюю благодарность и признательность компаниям «Модуль Электроник» и «Вентмашина» за предоставленное для работы над сатьёй оборудование.

Историческая справка.

Протокол Modbus был создан в 1979 году американской компанией Modicon (с 1996 года - структурная часть Schneider Electric SA), пионером в области производства программируемых логических контроллеров (*Programmable Logic Controller - PLC*). Разработка велась под руководством основателя компании Modicon Дика Морли (Dick Morley), которого часто называют "отцом PLC". Целью проекта было создание простого, открытого и надёжного способа связи между программируемыми контроллерами и периферией — датчиками, операторскими панелями, приводами и т.д..

После своего появления Modbus долгое время существовал как промышленный стандарт *де-факто.* В 2004 году Modbus был передан в управление некоммерческой организации Modbus Organization, Inc. и закрепился уже как *открытый отраслевой стандарт*.

С 2006 года Modbus кодифицирован *де-юре* и входит в структуру международного стандарта **IEC 61158**, наряду с другими промышленными протоколами (PROFIBUS, EtherCAT и др.). Сегодня Modbus используется по всему миру — от промышленности до автоматизации зданий (BMS, HVAC, учёт энергии), включая сценарии интеграции с KNX и с другими шинными технологиями.

Технические детали Modbus.

Modbus — коммуникационный протокол, работающий по принципу «ведущий-ведомый»: когда одно ведущее устройство (master) опрашивает одно или несколько ведомых устройств (slaves), либо отправляет им данные и команды.

Две наиболее часто используемые версии Modbus:

- **Modbus RTU** – компактная бинарная передача данных по последовательной линии RS-485 на витой паре;
- **Modbus TCP** - протокол, совместимый по формату данных с Modbus RTU, но адаптированный для сетей Ethernet (TCP/IP) и упрощающий подключение оборудования к OPC серверам, шлюзам, SCADA и серверным системам.

Дальнейшая информация относится к версии протокола Modbus RTU (Remote Terminal Unit).

Адресация устройств

В Modbus RTU максимальное возможное количество подключаемых к шине физических устройств 248, включая одно ведущее устройство (мастер). Ведомым устройствам назначаются уникальные адреса (slave ID) от 1 до 247.

Мастер шины Modbus своего адреса не имеет, поскольку ни одно ведомое устройство ему никогда по своей инициативе ничего не отправляет. В ответных пакетах, отправляемых ведомыми устройствами по запросам мастера, в поле адреса повторяются те же самые адреса (slave ID), на которые отправлялись запросы мастером.

Адрес ‘0’ – широковещательный (broadcast). В ответ на широковещательные сообщения мастера никакие ведомые устройства не отвечают.

Адреса 248-255 зарезервированы и официально не используются. Таблица ниже резюмирует сказанное:

Адрес (1 байт)	**Назначение**
0	Широковещательный адрес (broadcast)
1-247	Доступные slave-адреса
248-255	Резерв

Типы данных

Как упоминалось ранее, назначение Modbus - обеспечивать обмен данными контроллерного оборудования с периферийными устройствами ввода-вывода: кнопками, реле, датчиками физических величин, аналоговыми выходами и т.д. Это подразумевает двунаправленную передачу аналоговых и дискретных данных разного формата и назначения.

В соответствии с этой задачей в Modbus предусмотрен необходимый набор из четырех типов данных:

№	Название	Тип	Размер	Доступ	Назначение
1	Discrete Output Coils Дискретные выходы (Катушки)	DO	1 бит	Чтение / Запись	• Переключение реле • Чтение статусов реле
2	Discrete Input Contacts Дискретные входы	DI	1 бит	Чтение	• Бинарные входы для «сухих контактов» • Бинарные входы постоянного/переменного тока
3	Analog Input Registers Регистры ввода	AI	16 бит	Чтение	• Аналоговые входы • Измерения
4	Analog Output Holding Registers Регистры хранения	AO	16 бит	Чтение / Запись	• Передача и чтение параметров • Передача и чтение уставок • Передача команд

Адресация данных

Каждому из четырех типов данных Modbus выделен свой диапазон *логических адресов* (двухбайтные целые числа в диапазоне от 0 до 65535) в соответствии с таблицей ниже:

Тип данных	Логические адреса	Размер	Доступ
Coils (DO)	00001 - 09999	1 бит	Чтение / Запись
Discrete Inputs (DI)	10001 - 19999	1 бит	Чтение
Input Registers (AI)	30001 - 39999	16 бит	Чтение
Holding Registers (AO)	40001 - 49999	16 бит	Чтение / Запись

Таким образом, можно сказать, что пространство данных Modbus условно разделено по типам данных на 4 таблицы, с емкостью 9999 элементов в каждой таблице.

Заметим, что «круглые» адреса 10000, 20000, 30000, 40000 не входят в используемые диапазоны логических адресов. Кроме этого, диапазон 20001 – 29999 официальным протоколом Modbus также не используется (возможно, когда-то был зарезервирован компанией Modicon для проприетарных целей).

Описанная выше логическая адресация данных в Modbus носит презентационный характер и используется в технической документации для удобной идентификации регистров и однобитовых переменных на "человеческом" уровне и понимания того, к какому типу данных относится та или иная переменная или параметр.

Обращение к данным на физическом уровне протокола Modbus производится по *физическим адресам*. Для всех четырех типов данных Modbus диапазоны физических адресов совпадают и представляют собой последовательность двухбайтных целых чисел от 0 до 9998. Пересчет логического адреса в физический производится вычитанием фиксированного смещения. Для каждого типа данных смещение своё:

Тип данных	Логические адреса	Смещение	Физические адреса
Coils (DO)	00001 - 09999	00001	0 - 9998
Discrete Inputs (DI)	10001 - 19999	10001	0 - 9998
Input Registers (AI)	30001 - 39999	30001	0 - 9998
Holding Registers (AO)	40001 - 49999	40001	0 - 9998

Очевидно, физический адрес, передаваемый в сообщениях Modbus, уже не позволяет определить тип переменной, относящейся к сообщению – диапазоны физических адресов для всех типов данных одинаковы (см. таблицу выше). Поэтому тип запрашиваемой или передаваемой в сообщении переменной определяется только значением обязательного поля сообщения - '*Код функции* (*Function Code*)'.

Функции Modbus

Функция в Modbus — это передаваемый в каждом сообщении однобайтовый код (Function Code), который указывает смысл действия, которое ведущее (master) устройство хочет выполнить на ведомом (slave) устройстве. Такими действиями могут быть:

- чтение данных,
- запись значений,
- диагностика,
- сброс счётчиков,
- и другие.

Кроме вида действия (команды), функция определяет и тип данных, над которыми это действие выполняется. Таблица ниже содержит информацию об основных функциях Modbus:

Код (hex)	Тип данных	Действие	Размер
0x01	Coils (DO)	Чтение дискретных выходов	1 бит (R)
0x02	Discrete Inputs (DI)	Чтение дискретных входов	1 бит (R)
0x03	Holding Registers (AO)	Чтение регистров хранения	16 бит (R)
0x04	Input Registers (AI)	Чтение регистров ввода	16 бит (R)
0x05	Coils (DO)	Запись дискретных выходов	1 бит (W)
0x06	Holding Registers (AO)	Запись регистров хранения	16 бит (W)
0x0F	Coils (DO)	Запись нескольких дискретных выходов	1 бит (W)
0x10	Holding Registers (AO)	Запись нескольких регистров хранения	16 бит (W)

Кроме указанных в таблице часто используемых функций, протокол Modbus предусматривает также ряд диагностических и расширенных функций различного назначения. Информацию о том, какие функции протокола поддерживаются конкретными устройствами, их производители указывают в технических спецификациях.

В случае возникновения операционных ошибок, ведомые устройства Modbus передают мастеру в ответных сообщениях коды ошибок. При этом код функции в ответном сообщении заменяется на его сумму с числом 0x80. Сам однобайтовый код ошибки (Exception Code) передаётся в поле данных сообщения. Примеры ошибок Modbus:

Код ошибки	Тип ошибки
0x01	Неподдерживаемая функция (Illegal Function)
0x02	Недопустимый адрес регистра (Illegal Data Address)
0x03	Недопустимое значение (Illegal data value)
0x04	Внутренняя ошибка устройства (Slave Device Failure)
0x06	Устройство занято (Slave Device Busy)
0x0A	Ошибка шлюза (Gateway Path Unavailable)
0x0B	Устойство не отвечает (Gateway Target Device Failed to Respond)

Информационный обмен

В Modbus RTU информационный обмен представляет собой последовательность циклов "запрос мастера – ответ ведомого", идущих один за другим без наложений во времени. Мастер никогда не отправляет следующий запрос, пока не дождется ответа на предыдущий, либо в случае долгого отсутствия ожидаемого ответа не зафиксирует ошибку передачи по *таймауту*. Перекрывающихся сообщений или "параллельных" циклов обмена в Modbus RTU быть не может:

[M] — запрос к устройству 0x01
[S1] — ответ
[пауза]
[M] — запрос к устройству 0x02
[S2] — ответ
[пауза]
...

Каждое сообщение мастера требует ответа адресованного ведомого устройства. Если сообщение мастера не подразумевает ответной передачи каких-то данных, например, передаётся просто команда включения, то в ответ на него адресованное ведомое устройство лишь дублирует первоначальное сообщение мастера, подтверждая тем самым, что команда была успешно принята и обработана (*эхо-подтверждение*).

Ведомые устройства (slaves) никогда не инициируют передачу. Они всегда ждут запросов мастера и отвечают на них, если они адресованы им.

Таким образом, Modbus RTU - синхронный протокол, по которому мастер шины полностью контролирует обмен данными и командами.

Формат сообщений

На рисунке ниже представлен формат сообщений (кадров) Modbus RTU:

Первое поле размером 1 байт – это адрес назначения ведомого (slave) устройства в диапазоне 1-247. Следующий за адресом байт содержит код функции, определяющий операцию и тип данных к которым эта операция должна быть применена (например, 0x2 – чтение дискретного входа).

Третье поле сообщения Modbus RTU имеет переменную длинну и содержит передаваемые данные: например, адрес регистра или однобитовой дискретной переменной; количество данных; значения данных и пр..

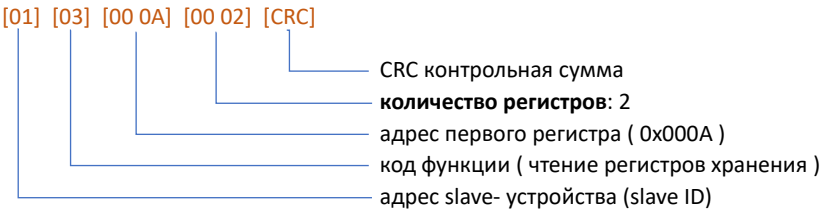
Замыкает сообщение двухбайтовая контрольная сумма CRC (*Cyclic Redundancy Check*) проверки целостности передаваемых данных.

Как указано выше, третье поле сообщения с данными имеет переменную длину. Для определения границы данных принимающее устройство использует код функции (по нему определяется тип данных) и явно указываемое количество экземпляров обрабатываемых данных (регистров, байтов или битов):

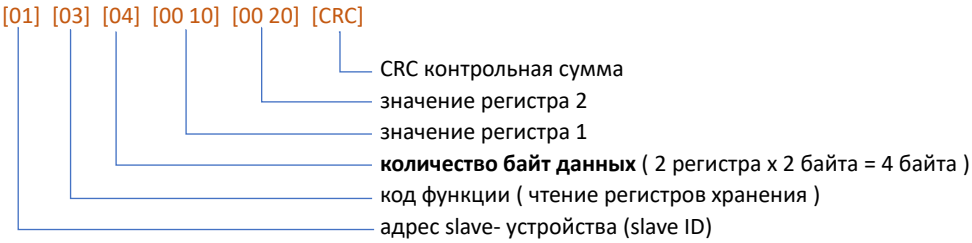
Функция	**Что обязательно указывается в поле длины данных**
0x01	Количество дискретных выходов | Coils (бит)
0x02	Количество дискретных входов | Discrete Inputs (бит)
0x03	Количество регистров хранения | Holding Registers (16-битных слов)
0x04	Количество регистров ввода | Input Registers (16-битных слов)

Пример 1: чтение двух регистров хранения (Holding Registers)

Запрос:



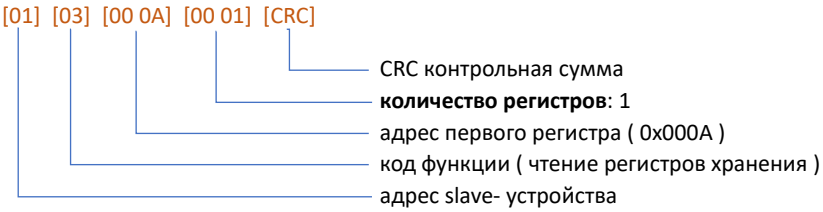
Ответ:



Блочный характер протокола Modbus диктует требование жесткого соблюдения формата сообщений, независимо от количества обрабатываемых данных. В частности, байтовое поле длины запрашиваемых или передаваемых данных всегда явно присутствует в сообщениях, даже если речь идет об единичном экземпляре данных (Пример 2).

Пример 2: чтение одного регистра хранения (Holding Registers)

Запрос:



Кухня интеграции.

Интеграция систем, построенных на различных шинных протоколах, например, KNX и Modbus, требует ясного понимания роли аппаратного шлюза как посредника между двумя разнородными средами обмена данными. Шлюз обеспечивает прозрачную двунаправленную передачу управляющих команд и состояний, выступая в качестве «переводчика» между точками управления в каждой системе.

Далее описывается поэтапный процесс интеграции двух систем в конткесте Modbus и KNX протоколов.

Определение точек управления.

Первым шагом в процессе интеграции всегда является формирование списка точек управления (control data points) — дискретных и аналоговых сигналов, которые должны быть доступны для обеих систем. Это могут быть команды включения/выключения, задания уставок, показания датчиков и т.д.

В KNX системе точками управления являются *групповые объекты,* коммуницирущие через шину KNX с помощью телеграмм, отправляемых на связанные с ними *групповые адреса*.

Точки управления в Modbus – это *регистры* и *дискретные переменные*, адресуемые двумя адресными компонентами: адресом ведомого устройства (slave ID = 1..247) и физическим адресом регистра или дискретной переменной в устройстве (0..9998).

Маппинг сигналов.

На основе перечней точек управления выполняется маппинг — сопоставление конкретных регистров и адресов Modbus с групповыми объектами и групповыми адресами KNX. На этом этапе важно учитывать как физические структуры сетей, так и логику работы приложений, чтобы интеграция работала корректно, без задержек, конфликтов и логических ошибок.

Преобразование типов данных.

KNX и Modbus используют разные форматы представления данных: от простых булевых значений до чисел с плавающей точкой и кодировок температур, времени суток или статусов. Шлюз должен выполнять корректное преобразование данных разных типов (data type conversion), сохраняя точность и физический смысл значений.

В KNX используемые форматы типов данных жестко определены на уровне системной спецификации “Interworking. Datapoints Types” стандарта KNX. В Modbus типы данных (Coils, Discrete Inputs, Holding Registers, Input Registers) не определяют кодирование данных, а лишь их размер: это двухбайтный регистр или однобитовая переменная. Кодировка и интерпретация данных в Modbus определяются документацией производителя.

В Modbus нет универсальной системы кодификации данных, подобной DPTs в KNX — формат и смысл данных в регистрах всегда нужно смотреть в маппинге (*Modbus Map*) или таблице регистров (*Register Table*) конкретного устройства, предоставляемых его производителем.

В таблице ниже указаны типовые DPTs из KNX и их наиболее близкие эквиваленты в Modbus:

KNX				Modbus эквивалент	Комментарий по интеграции
DPTs	**Описание**	**Формат**	**Кодировка**		
1.xxx	DPT_Switch DPT_Bool …	1 бит [0,1]	Выкл/Вкл Ложь/Истина	Coils или Discrete Inputs	Coils – запись Discrete Inputs - чтение
5.001	DPT_Scaling	8 бит : U_8 [0..255]	0..100%	Holding Register 16 бит: U_{16}	Значение размещается в младшем байте регистра
5.004	DPT_Percent	8 бит : U_8 [0..255]	0..255%	Holding Register 16 бит: U_{16}	Значение размещается в младшем байте регистра
7.xxx	DPT_Value_2_Ucount …	16 бит: U_{16} [0..65 535]	Импульсы …	Holding Register 16 бит: U_{16}	Формат совпадает, прямая передача
8.xxx	DPT_Value_2_Count …	16 бит: V_{16} [-32 768…+32 767]	Импульсы …	Holding Register 16 бит: V_{16}	Знаковое представление в формате Int16
9.xxx	DPT_Value_4_Float	16 бит: F_{16} 4 бит: экспонента [0…15] 12 бит: мантисса [-2 048 … 2 047] (двоичное дополнение) диапазон: [-671 088,64 … 670 760,96]	Значения физических величин с точностью 0.01	Holding Register 16 бит: F_{16}	Требуется преобразование KNX float ↔ IEEE754
10.001	DPT_TimeOfDay	24 бит: $N_3U_5r_2U_6r_2U_6$ $b_{23}..b_{21}$: день [1..7] $b_{20}..b_{16}$: часы [0..23] $b_{13}..b_8$: мин [1..59] $b_5..b_0$: сек [1..59]	День, часы, минуты, секунды	2 Holding Registers 32 бит: U_{32}	Требуется сериализация полей или их раздельное хранение в четырех регистрах
11.001	DPT_Date	24 бит: $r_3U_5r_4U_4r_1U_7$ $b_{20}..b_{16}$: день [1..31] $b_{11}..b_8$: месяц [1..12] $b_6..b_0$: год [0..99]	День месяца, месяц, год (≥ 90: 20 век; <90: 21 век)	2 Holding Registers 32 бит: U_{32}	Требуется сериализация полей или их раздельное хранение в трех регистрах
12.001	DPT_Value_4_Ucount	32 бит: U_{32} [0…4 294 967 295]	Импульсы …	2 Holding Registers 32 бит: U_{32}	Прямое соответствие, при совпадении порядка байт/слов
13.xxx	DPT_Value_4_Vcount …	32 бит: V_{32} [-2 147 483 648 … 2 147 483 647]	Импульсы …	2 Holding Registers 32 бит: V_{32}	Прямое соответствие, при совпадении порядка байт/слов
14.xxx	DPT_Value_4_Float	32 бит: F_{32} b_{31}: знак {0,1} $b_{30}..b_{23}$: экспонента [0…255] $b_{22}..b_0$: мантисса [0…8 388 607]	Значения физических величин с точностью 1	2 Holding Registers 32 бит: F_{32}	Прямое соответствие, при совпадении порядка байт/слов и с кодировкой IEEE754
18.001	DPT_SceneControl	8 бит: $B_1r_1U_6$ b_7: 0- вызов сцены 1- запись сцены $b_5..b_0$: номер сцены [0..63]	Команда вызова или записи сцены и номер сцены	Holding Register 16 бит: U_{16}	Часто используется битовая маска для выделения полей
232.600	DPT_Colour_RGB	24 бит: U_8 U_8 U_8 $b_{23}..b_{16}$: R [0..255] $b_{15}..b_8$: G [0..255] $b_7..b_0$: B [0..255]	Цветовые RGB компонеты в двоичной кодировке	2 Holding Registers 32 бит: U_{32}	Требуется сериализация полей или их раздельное хранение в трех регистрах; возможно преобразование в hex-формат

Логическая согласованность.

При интеграции KNX и Modbus важно обеспечить логическую согласованность при двустороннем обмене данными. Даже в случае физической и протокольной совместимости, расхождение в логике обработки команд может привести к ошибкам в управлении или некорректному отображению состояний. Например, изменение состояния на стороне KNX должно своевременно и без искажений отражаться в системе Modbus, и наоборот. Это особенно критично в системах, где управление

осуществляется как вручную, так и автоматически, и где статус объекта управления должен точно отображаться на обеих сторонах.

Для этого применяются механизмы буферизации и фильтрации. Буферизация позволяет временно сохранять поступающие данные перед их обработкой, что предотвращает конфликты при одновременном изменении параметров с обеих сторон. Фильтрация сообщений позволяет отбрасывать избыточные или нерелевантные телеграммы, например: повторяющиеся значения, значения без изменений, слишком частые обновления или колебания сигнала (дребезг). Также может применяться пороговая фильтрация — передача аналогового значения только при изменении, превышающем заданный процент или шаг.

Особое внимание следует уделять приоритетности команд в ситуациях, когда управляющие действия могут поступать одновременно с разных сторон. Например, система Modbus может инициировать включение освещения по расписанию, в то время как пользователь пытается управлять тем же светом вручную через панель KNX.

В таких случаях необходимо определить правила приоритета обработки команд, например:

- приоритет за ручным управлением — Modbus-команды временно игнорируются;
- приоритет за автоматикой — команды от KNX не передаются в Modbus до истечения тайм-аута;
- блокировка одной стороны управления в заданные периоды времени (например, запрет ручного управления в ночное время).

Таким образом, обеспечение логической согласованности требует не только правильной адресации и форматов передаваемых данных, но и продуманной обработки событий: с буферизацией, фильтрацией и гибким управлением приоритетами. Это позволяет исключить логические конфликты и добиться стабильной, предсказуемой работы интегрированной системы.

Настройка и тестирование.

Эффективная интеграция требует внимательной настройки параметров шлюза: скоростей обмена, таймаутов, последовательности опроса регистров. После настройки проводится тестирование каждого канала — как в направлении от KNX к Modbus, так и в обратном, — с контролем времени реакции и корректности данных.

Документирование.

Полное и подробное документирование маппинга, форматов данных и логики взаимодействия интегрируемых систем обязательно для дальнейшего сопровождения инсталляции и исключения ошибок при ее модернизации. Иллюстрации, отражающие архитектуру шлюза, последовательность преобразования данных и таблицы маппинга, помогают интегратору быстрее ориентироваться в структуре проекта и сокращают время пусконаладочных работ.

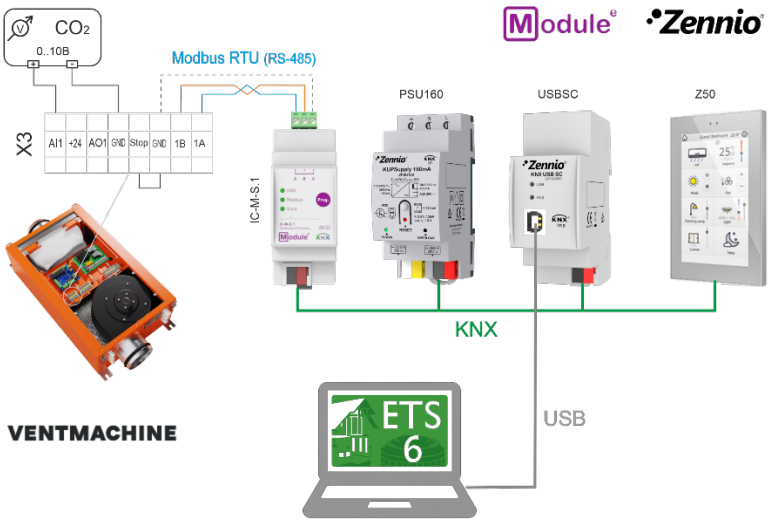
KNX интеграция ПВУ Ventmachine Orange 350 GTC.

Оборудование.

В предыдущей части статьи были рассмотрены основы протокола Modbus и общая методология межсистемной интеграции. Далее описывается интеграция сенсорной KNX панели Z50 («Zennio», Испания) и ПВУ Ventmachine Orange 350 GTC («Вентмашина», Россия). Предполагается, что панель Z50 выпоняет роль средства пользовательского управления и мониторинга.

ПВУ компании «Вентмашина» оснащены Modbus коммуникацией в качестве штатного средства подключения к внешнему управлению (например, BMS). Поэтому в качестве интерфейсного устройства между вентустановкой и KNX шиной был выбран шлюз IC-M-S.1 KNX-Modbus от компании «Модуль Электроник», Россия.

Реализованная схема подключения оборудования изображена на рисунке ниже:



Также в инсталляции использовались системные устройства Zennio: источник питания KUPSupply 160mA и интерфейс данных KNX USB SC. В обработке ошибок ПВУ был задействован программный модуль логических функций актуатора Zennio MINiBOX-45v3.

В ПВУ Ventmachine Orange 350 для контроля качества воздуха предусмотрена возможность подключения внешнего сенсора CO_2 с аналоговым выходом 0-10В. Для имитации такого сенсора в инсталляции использовался стандартный ИП постоянного тока с регулируемым выходом 0-30В.

Точки управления Modbus.

В соответствии с общим подходом к процессу интеграции вначале необходимо определить перечень интегрируемых точек управления. В данном случае точки управления представляют собой описанные производителем в технической документации рабочие параметры ПВУ, доступные через Modbus коммуникацию. В демонстрационных целях из общего обширного списка Modbus регистров ПВУ для интеграции с KNX были выбраны следующие:

№пп	Точка управления Modbus	Формат данных Modbus	Адрес регистра hex (dec)
1	Включение и выключение ПВУ	Holding Register: U_{16} $b_{15}..b_1$: резерв b_0: { вкл (1) , выкл (0) }	0x0002 (2)

2	Скорость вентилятора притока ПВУ (режим управления 0-10В)	Holding Register: U_{16} $b_{15}..b_4$: резерв $b_3..b_0$: { 1, …, 10 }	0x0020 (32)
3	Уставка температуры воздуха в канале притока ПВУ (в 0.1 °C)	Holding Register: U_{16} [150 … 300]	0x001F (31)
4	Температура воздуха в канале притока ПВУ (в 0.1 °C)	Input Register: V_{16}	0x0009 (9)
5	Относительная влажность воздуха в помещении (в %)	Input Register: U_{16}	0x003A (58)
6	Код текущей ошибки	Input Register: U_{16} b_0 : 1 – неисправность в цепи датчика температуры в канале b_9 : 1 – неисправность в цепи датчика давления вентилятора b_{10} : 1 – авария вентилятора b_{11} : 1 – пожар b_{13} : 1 – перегрев калорифера	0x0004 (4)
7	Скорость вентилятора вытяжки ПВУ (режим управления 0-10В)	Input Register: U_{16} $b_{15}..b_4$: резерв $b_3..b_0$: { 1, …, 10 }	0x001E (30)
8	Концентрация CO_2 в помещении, скорректированная (в ppm)	Input Register: U_{16}	0x003D (61)

Точки управления KNX.

Для ввода и отображения значений перечисленных выше параметров ПВУ на стороне KNX также должны быть предусмотрены соответствующие точки управления. В нашем случае такими точками управления являются групповые объекты элементов визуального управления (controls) панели Zennio Z50 (см. таблицу ниже). Элементы управления размещаются на двух пользовательских страницах панели Z50.

На *первой странице* панели размещаются элементы управления и мониторинга ПВУ:

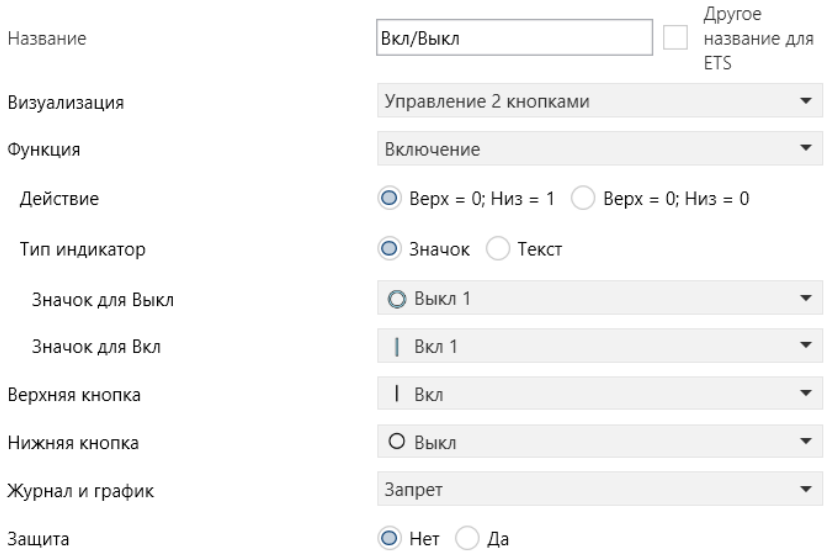
- Включение и выключение ПВУ (бинарный переключатель)
- Скорость приточного вентилятора (дискретный регулятор)
- Уставка температуры (температурный регулятор в °C)
- Относительная влажность воздуха в % (числовой индикатор)
- Скорость вытяжного вентилятора (числовой индикатор)
- Концентрация CO_2 (числовой индикатор)
- Индикатор неисправности ПВУ (графический индикатор)

На *второй странице* панели размещаются отдельные графические индикаторы для пяти видов неисправностей ПВУ и аварийных ситуаций:

- Неисправность канального датчика температуры воздуха T1
- Неисправность датчика давления приточного вентилятора
- Неисправность приточного вентилятора
- Пожар
- Перегрев калорифера

№ Управления (controls) панели Z50 и их настройка

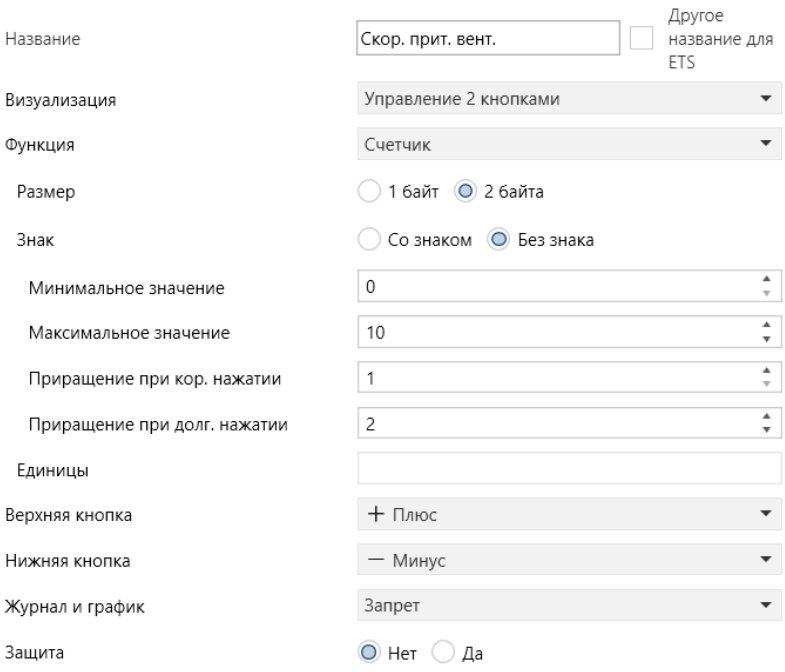
1. Включение/Выключение ПВУ



Групповые объекты:

#601 - [C1] [Вкл/Выкл ПВУ] Переключатель -Управление	DPT 1.001 Switch
#600 - [C1] [Вкл/Выкл ПВУ] Переключатель - Статус	DPT 1.001 Switch

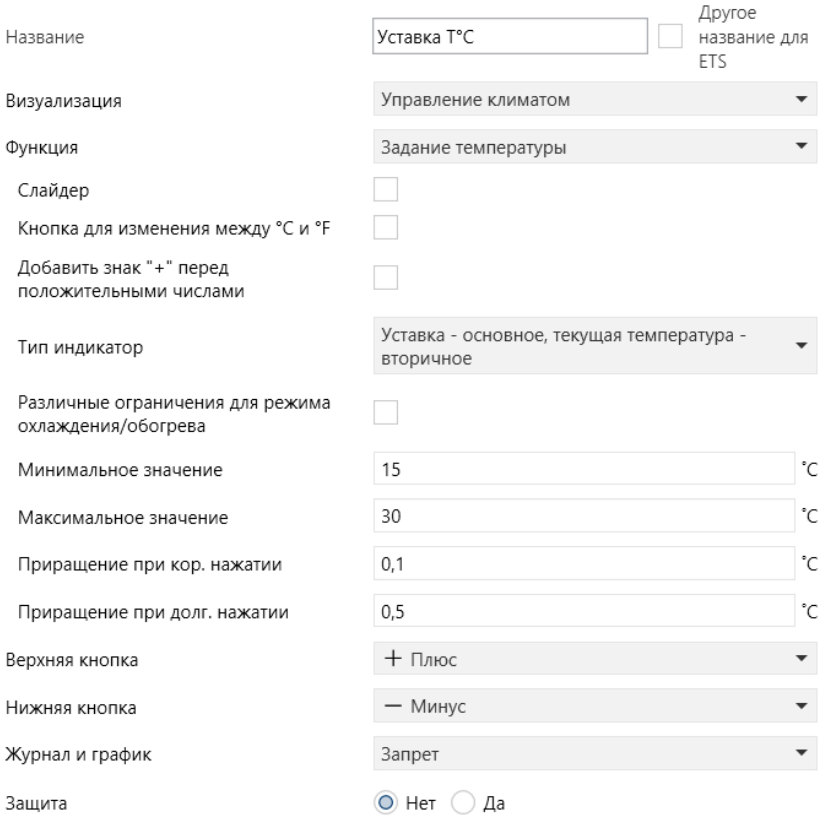
2. Скорость приточного вентилятора



Групповые объекты:

#607 - [C2] [Скорость вент.] Целое число 2-байт беззнак. управление	DPT 7.001 Pulses
#606 - [C2] [Скорость вент.] Целое число 2-байт беззнак. статус	DPT 7.001 Pulses

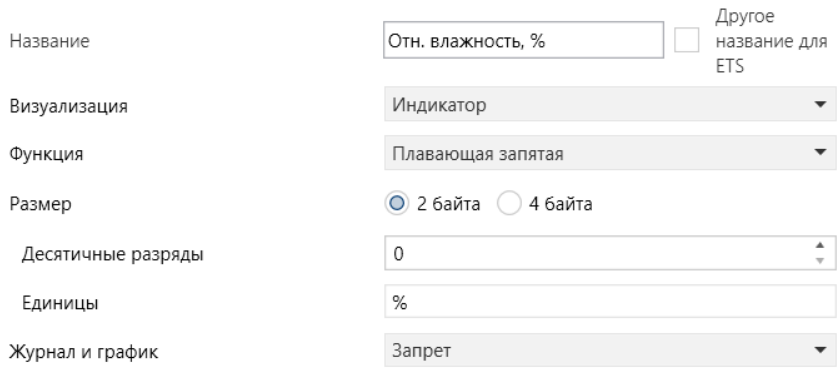
3. Уставка температуры приточного воздуха в °C



Групповые объекты:

#613 - [C3] [Уставка темп. °C] Уставка -Управление температурой	DPT 9.001 Temperature °C
#612 - [C3] [Уставка темп. °C] Уставка -Температурный статус	DPT 9.001 Temperature °C
#615 - [C3] [Уставка темп. °C] Текущая температура статус	DPT 9.001 Temperature °C

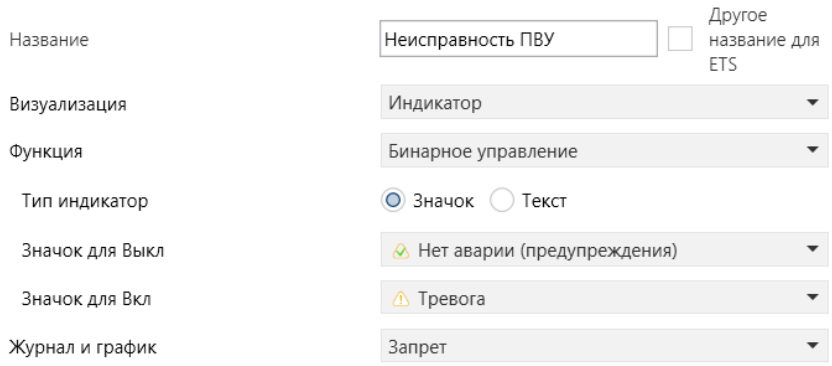
4. Относительная влажность воздуха, %



Групповые объекты:

#618 - [C4] [Отн. влажность, %] Плав. 2-байтовый статус	DPT 9.007 Humidity %

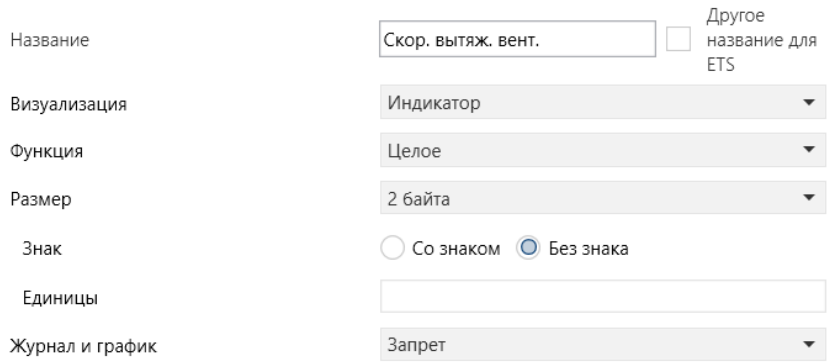
5. Неисправность ПВУ (общий сигнал)



Групповые объекты:

#624 - [C5] [Неисправность ПВУ] Переключатель - Статус	DPT 1.001 Switch

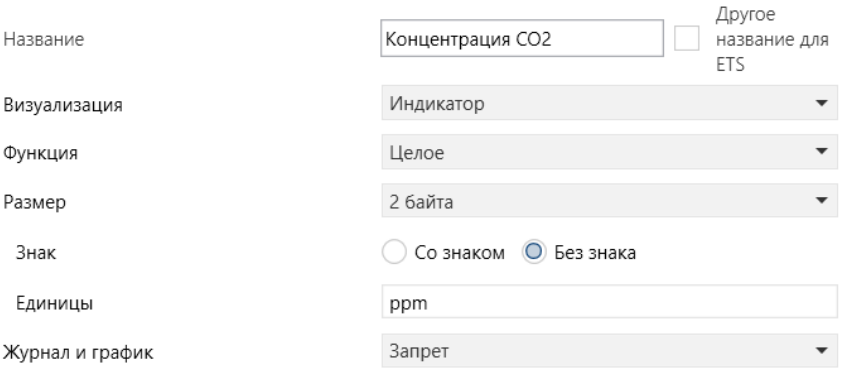
6. Скорость вытяжного вентилятора



Групповые объекты:

#630 - [C6] [Скорость вытяжка] Целое число 2-байт беззнак. статус	DPT 7.001 Pulses

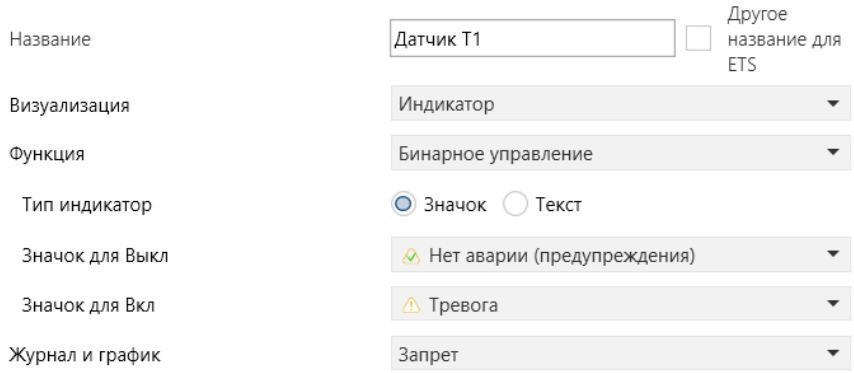
7. Концентрация CO_2 в ppm



Групповые объекты:

#636 - [C7] [Концетр.CO2] Целое число 2-байт беззнак. статус	DPT 7.001 Pulses

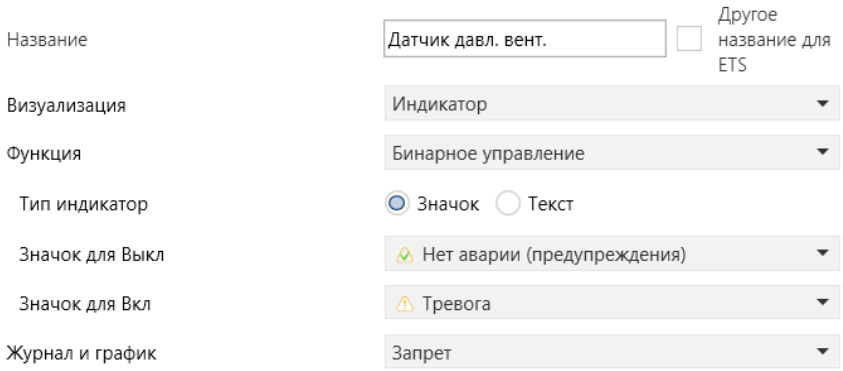
8. Датчик температуры DT1 - Ошибка



Групповые объекты:

#642 - [C8] [Датчик ДТ1] Переключатель - Статус	DPT 1.001 Switch

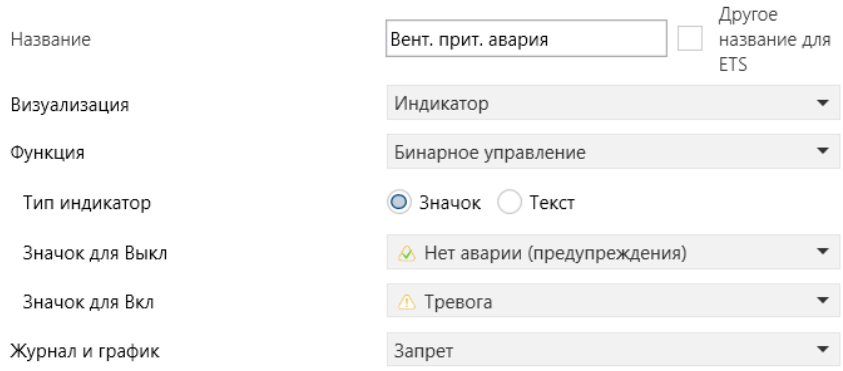
9. Датчик давления вентилятора - Ошибка



Групповые объекты:

#648 - [C9] [ДД вентилятор] Переключатель - Статус	DPT 1.001 Switch

10. Вентилятор приточный - Авария



Групповые объекты:

#654 - [C10] [Вентилятор авария] Переключатель - Статус	DPT 1.001 Switch

11. Пожар

Название	Пожар	Другое название для ETS
Визуализация	Индикатор	
Функция	Бинарное управление	
Тип индикатор	Значок / Текст	
Значок для Выкл	Нет аварии (предупреждения)	
Значок для Вкл	Предупреждение - дым	
Журнал и график	Запрет	

Групповые объекты:

#660 - [C11] [Пожар] Переключатель - Статус DPT 1.001 Switch

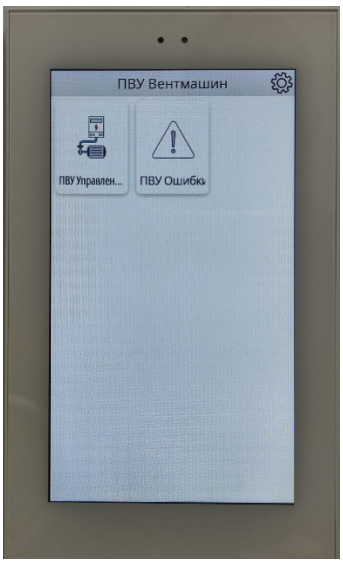
12. Перегрев калорифера

Название	Перегрев калорифер	Другое название для ETS
Визуализация	Индикатор	
Функция	Бинарное управление	
Тип индикатор	Значок / Текст	
Значок для Выкл	Нет аварии (предупреждения)	
Значок для Вкл	Тревога	
Журнал и график	Запрет	

Групповые объекты:

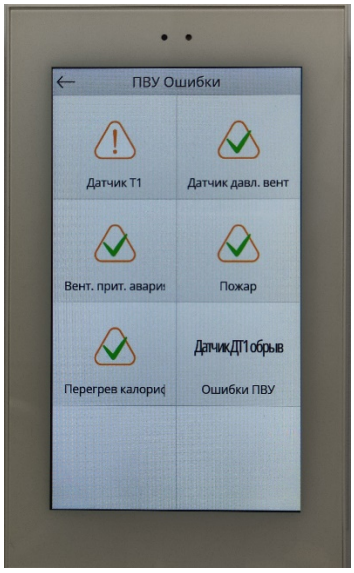
#666 - [C12] [Перегрев калорифера] Переключатель - Статус DPT 1.001 Switch

Следующие фотографии отображают интерфейс, получившийся в результате произведенных настроек панели Z50:



Меню

Страница 1



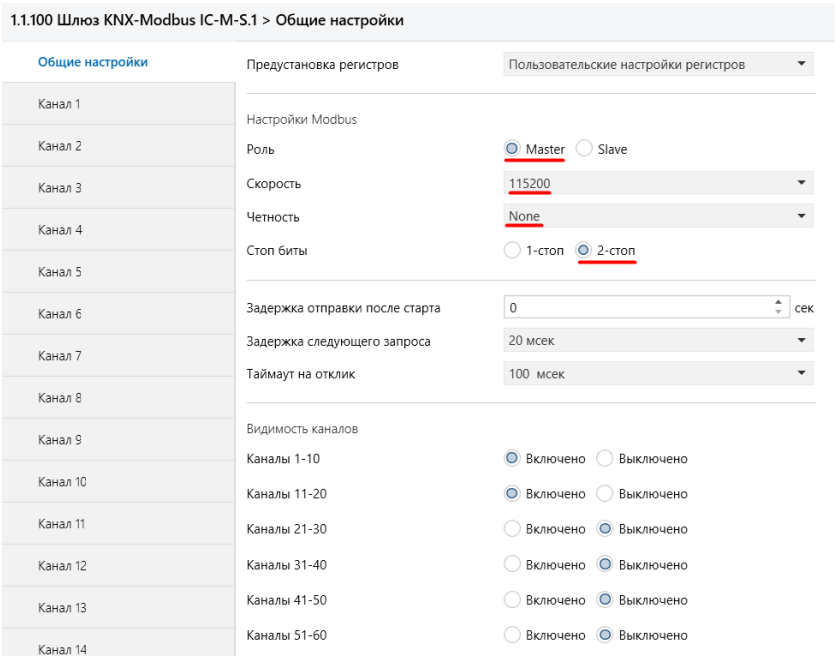
Страница 2

Настройка шлюза IC-M-S.1 KNX-Modbus компании «Модуль Электроник» полностью производится в программе ETS.

Для корректного связывания шлюза с шиной Modbus необходимо выполнить настройку общих параметров, определяющих его работу со стороны Modbus:

- Роль шлюза: Мастер (Master) или Ведомый (Slave)
- Скорость передачи: от 1200 до 115200 бит/с
- Контроль ошибок: отсутствует (None), четная сумма (Even), нечетная сумма (Odd)
- Количество стоп-бит: 1 или 2.

Данные для этих настроек берутся из технической документации подключаемого Modbus оборудования. ПВУ Ventmachine Orange требует следующих настроек:



От выбора роли шлюза Master или Slave зависит будет ли он опрашивать регистры ведомых устройств, или сам будет являться источником регистров для мастера шины. В случае, если сторона KNX является основной BMS в инсталляции, то шлюз выполняет роль мастера.

Заметим, что при выборе роли шлюза как ведомого устройства в блоке ‘Настройки Modbus’ появляется поле “Адрес Slave” для ввода Modbus адреса шлюза (1…247).

На вкладке “Общие настройки” также определяется какое количество каналов шлюза предполагается использовать. В нашем случае из максимально возможных 60 каналов активировано 20 каналов.

Настройка каналов шлюза (маппинг).

В процессе настройки каналов шлюза производится попарное сопоставление и связывание точек управления интегрируемых систем (маппинг). Каждая пара связываемых переменных занимает свой отдельный канал.

Настройка канала шлюза в среде ETS определяет его целевой Mobus адрес (Slave ID и адрес регистра) и преобразование типов данных связываемых в этом канале точек управления.

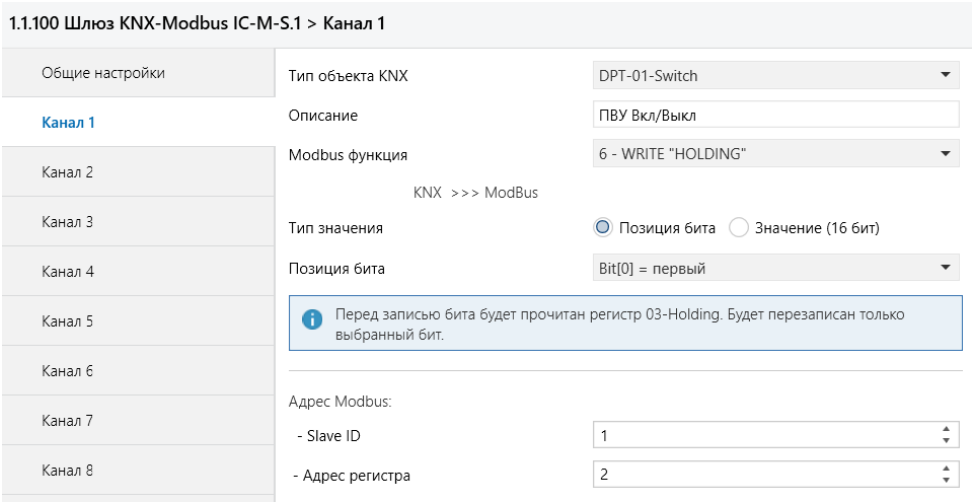
Для каждого канала шлюза устанавливаются параметры:

- *Тип данных объекта KNX* (DPTs):
  DPT-01-Switch
  DPT-03-Dimming
  DPT-05-Percent
  DPT-06-Signed-8
  DPT-08-Word-16
  DPT-09-Float
- *Функция Modbus*:
  1-READ «COIL»
  2-READ «DISCRET»
  3-READ «HOLDING»
  4-READ «INPUT»
  5-WRITE «COIL»
  6-WRITE «HOLDING»
  Значение параметра должно соответствовать желаемому направлению передачи и типу данных целевой переменной Modbus.
- *Внутренние преобразования и диапазоны*, действующие при передаче данных из одного протокола в другой. Состав этих параметров зависит от выбора первых двух настроек – типа данных объекта KNX и функции Modbus (см. руководство пользователя шлюза IC-M-S.1 KNX-Modbus).
- *Адрес Modbus*, состоящий из Slave ID и адреса регистра (привязка канала к переменной Modbus). Заметим, что привязка канала к точке управления KNX производится стандартным групповым связыванием в ETS группового объекта канала шлюза в ETS с групповым объектом точки управления KNX.

Ниже в таблице приводятся настройки использованных в инсталляции каналов.

Канал # Настройка канала шлюза IC-M-S.1

1 Включение/Выключение ПВУ - управление



Для включения ПВУ из KNX достаточно 1-битовой команды. Значит, значение параметра ‘Тип объекта KNX’ должно быть DPT-01-Switch.

Сама команда включения должна передаваться по направлению из KNX в Modbus, то есть это должна быть команда WRITE.

Из заводского описания регистров ПВУ следует, что включение и выключение вентиляционной установки производится через младший бит b(0) регистра хранения (Holding Register), с адресом 0x0002 (2). Также из документации ПВУ узнаем, что её контроллер имеет адрес устройства Slave ID равный 1.

На основании перечисленной информации производим следующие настройки:

- ✓ Modbus функция -> 6-WRITE “HOLDING”
- ✓ Тип значения -> Позиция бита Bit(0) = первый

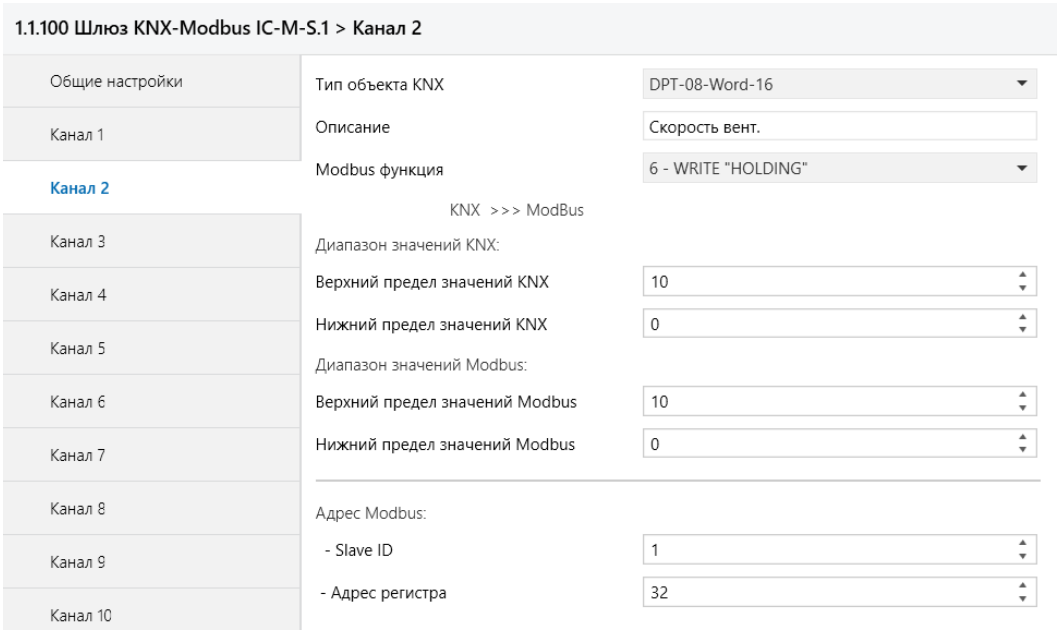
Адрес Modbus

- ✓ Slave ID -> 1
- ✓ Адрес регистра -> 2

На этом настройка канала #1 заканчивается.

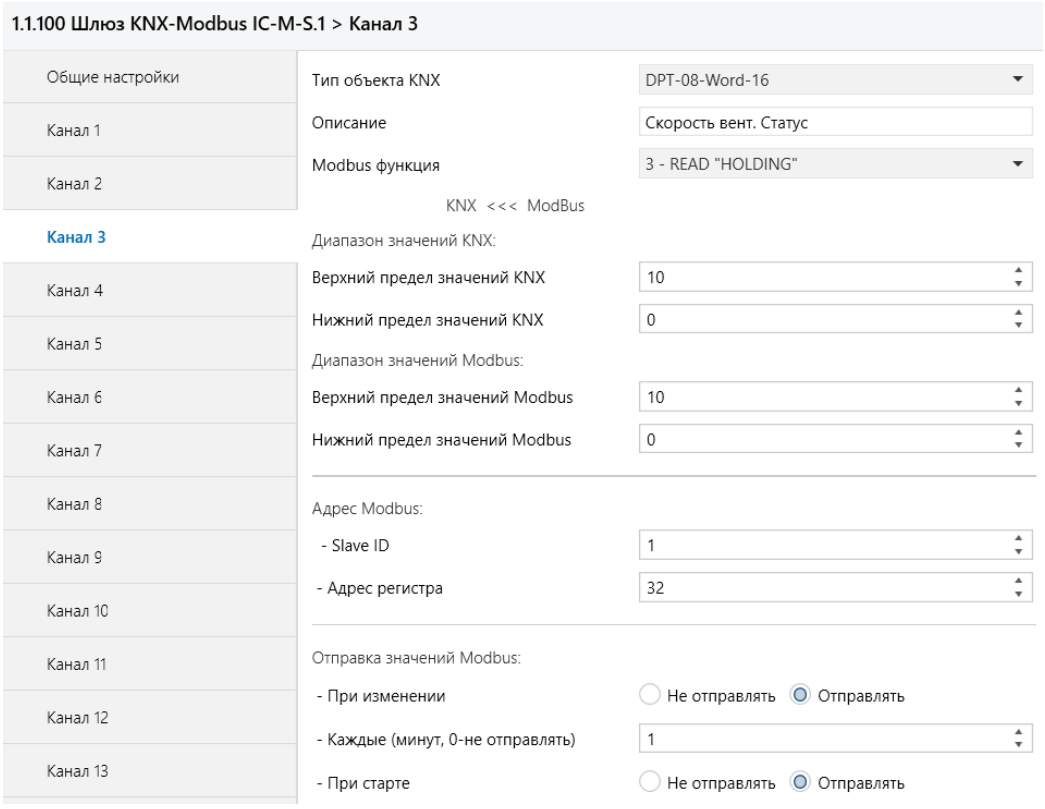
При поступлении в шлюз команды включения (или выключения) ПВУ, шлюз считывает из регистра Modbus с адресом #2 устройства #1 его текущее значение, устанвливает младший бит b(0) в значение 1 (или 0) и записывает полученный резльтат обратно в регистр #2.

2 Скорость приточного вентилятора - управление



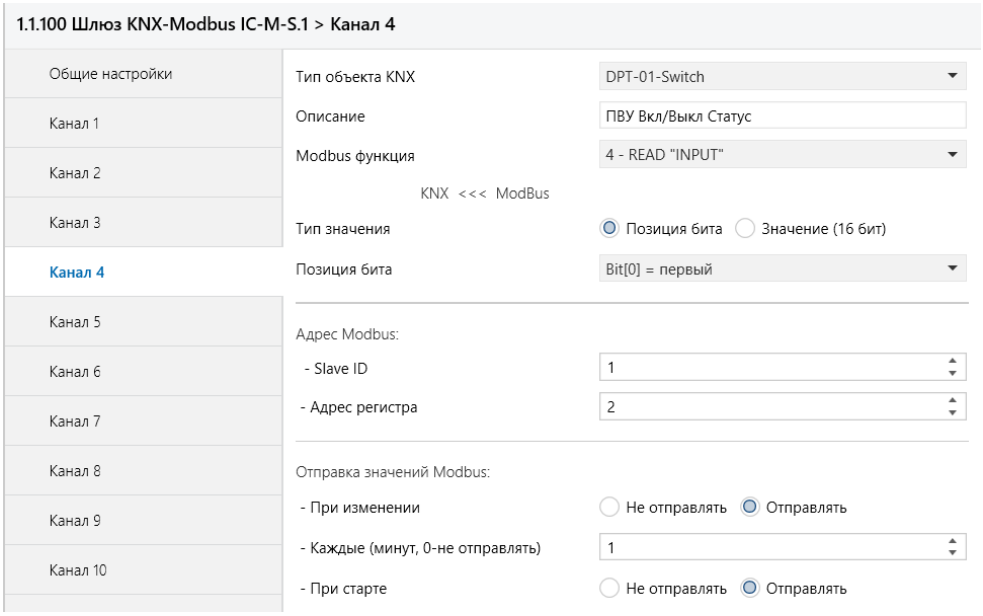
В соответствии с документацией ПВУ Ventmachine скорость приточного вентилятора задается целым положительным числом в диапазоне от 1 до 10 (в режиме с аналоговым 0-10В управлением вентилятором).

3 Скорость приточного вентилятора - статус



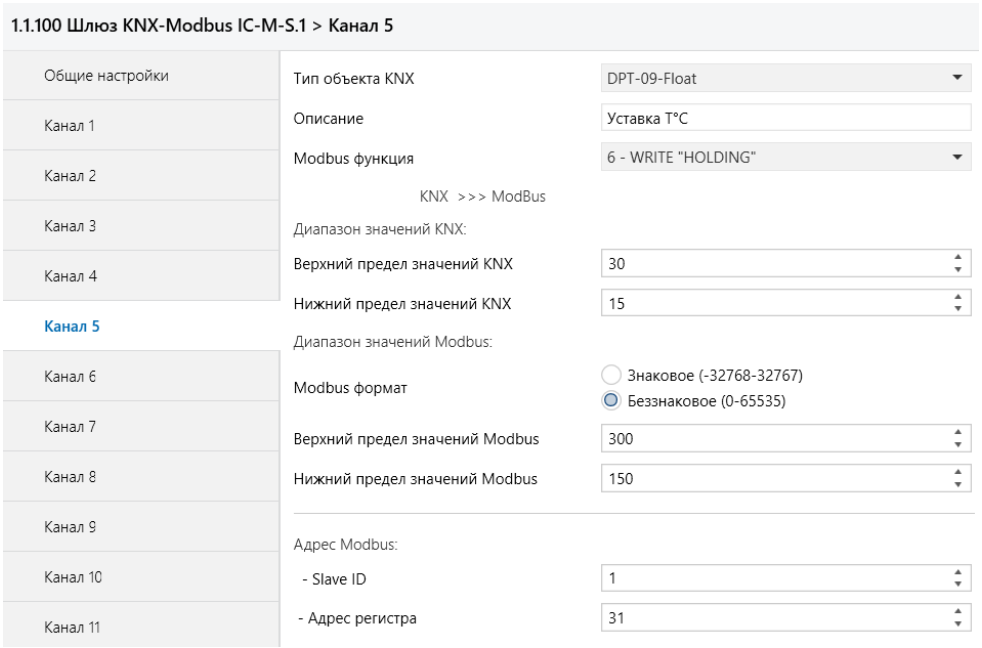
Настройка канала 3 аналогична настройкам канала 2, за исключением выбора функции – в данном случае передача данных должна производится в противоположном направлении, то есть выбирается функция чтения регистра хранения (3- READ “HOLDING”), вместо записи (6-WRITE “HOLDING”).

4 Включение/Выключение ПВУ – статус



Статус включения ПВУ, согласно документации производителя, отображается первым битом регистра ввода (Input Register) с адресом 0x0002 (2). Для его чтения используется функция 4-READ “INPUT”. Со стороны KNX для отображения бинарного статуса включения достаточно типа DPT-01-Switch. С учетом этого в настройках преобразования данных устанавливается позиционное выделение первого бита (bit[0]).

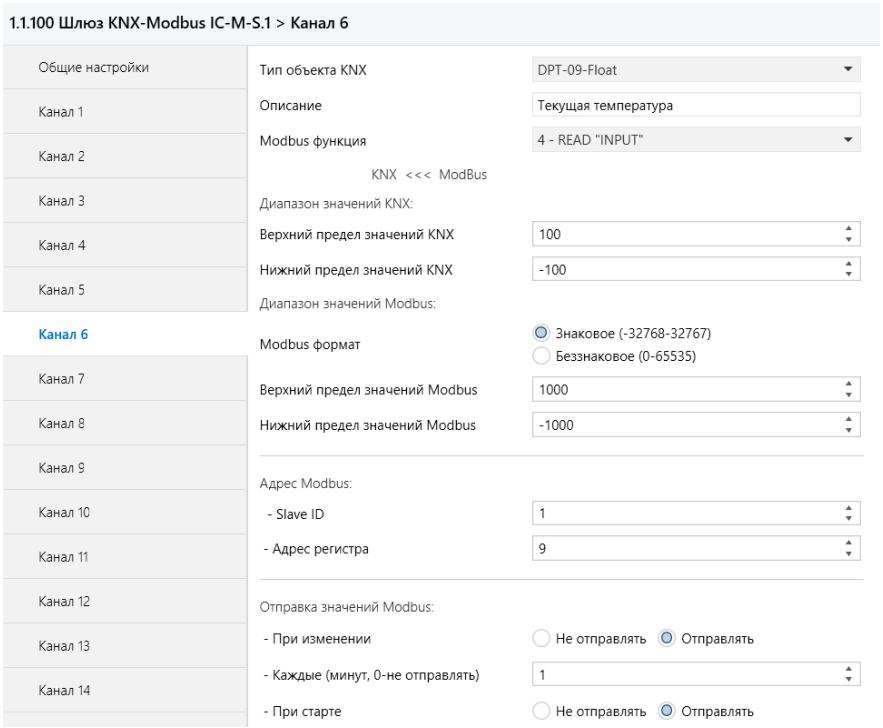
5 Уставка температуры приточного воздуха в °C - управление



В документации ПВУ указано, что уставка температуры приточного воздуха содержится в регистре хранения (Holding Register) #31 и устанавливается в диапазоне от 15 до 30 °C с использованием беззнакового целого 2-байтного числа в десятых долях градуса.

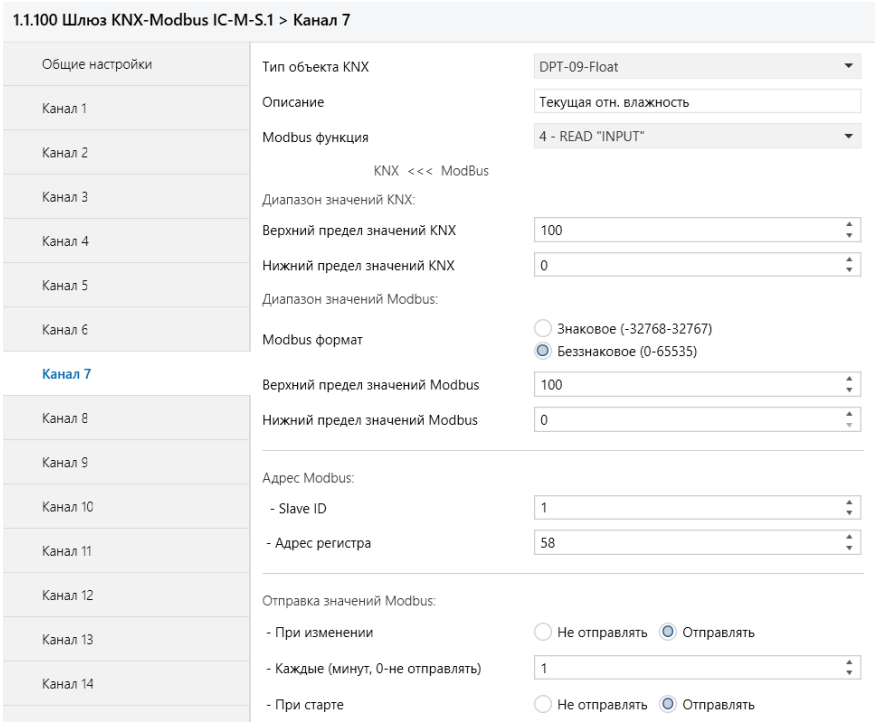
Таким образом, диапазон значений KNX для DPT-09-Float устанавливается от 15 до 30, а для Modbus – от 150 до 300.

6 Текущая температура воздуха °C



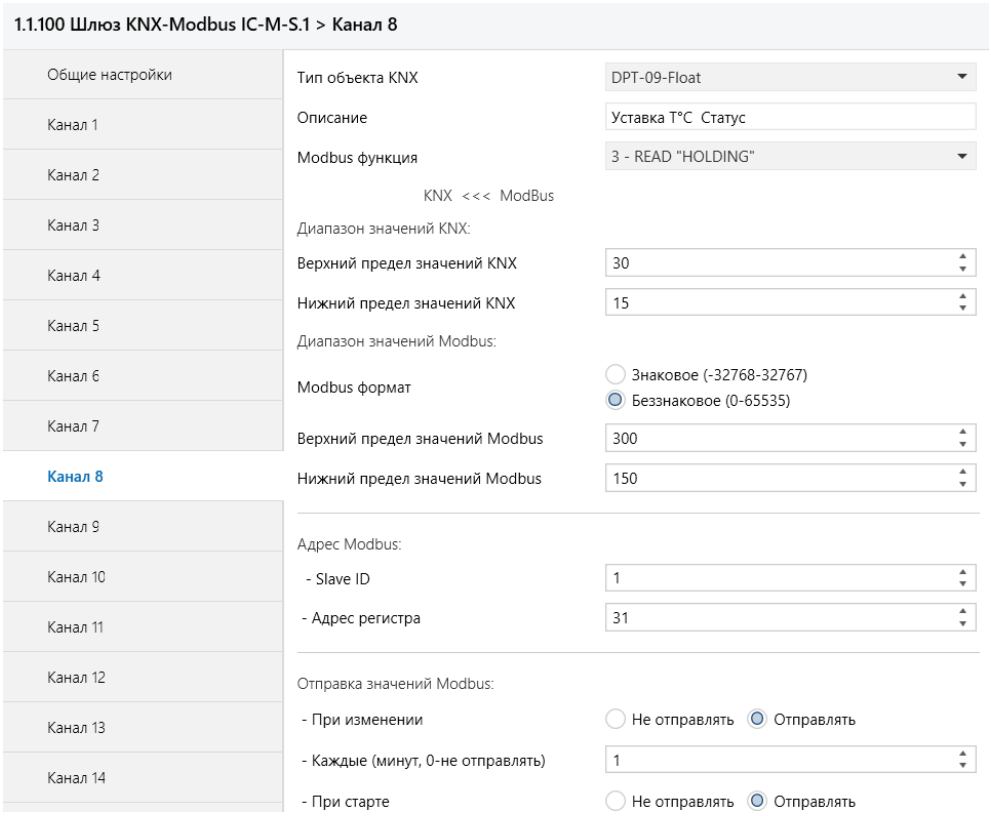
Согласно документации ПВУ, скорректированная температура воздуха в приточном канале содержится в регистре ввода #09 в формате знакового целого числа в десятых долях °C. Таким образом, диапазон значений KNX для DPT-09-Float устанавливается от -100 до 100, а для Modbus, соответственно от -1000 до 1000.

7 Текущая относительная влажность воздуха %



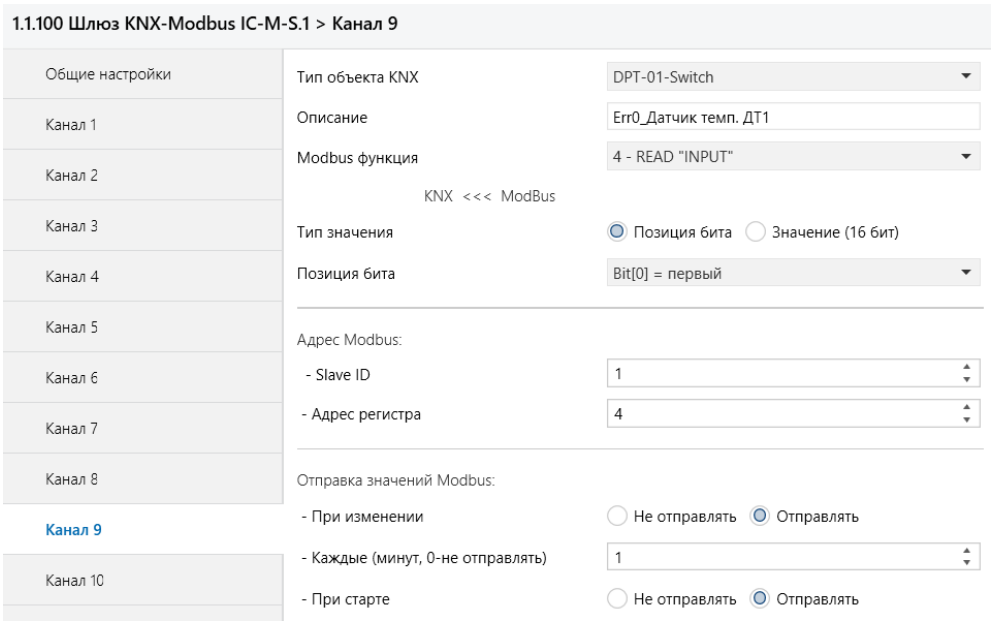
Относительная влажность воздуха в % содержится в регистре ввода #58 в формате беззнакового целого числа. Таким образом, диапазон значений KNX для DPT-09-Float и для Modbus устанавливаем от 0 до 100.

8 Уставка температуры приточного воздуха в °C - статус



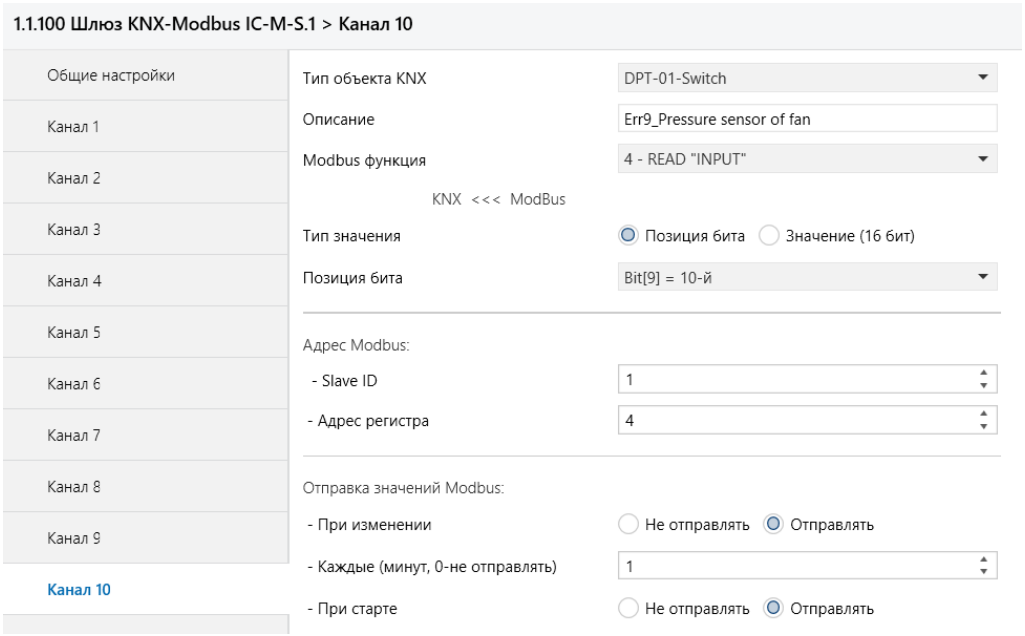
Настройка канала аналогична настройке канала 5, но с противоположным направлением передачи данных - в этом случае используется функция READ вместо функции WRITE.

9 Датчик температуры T1 - Ошибка



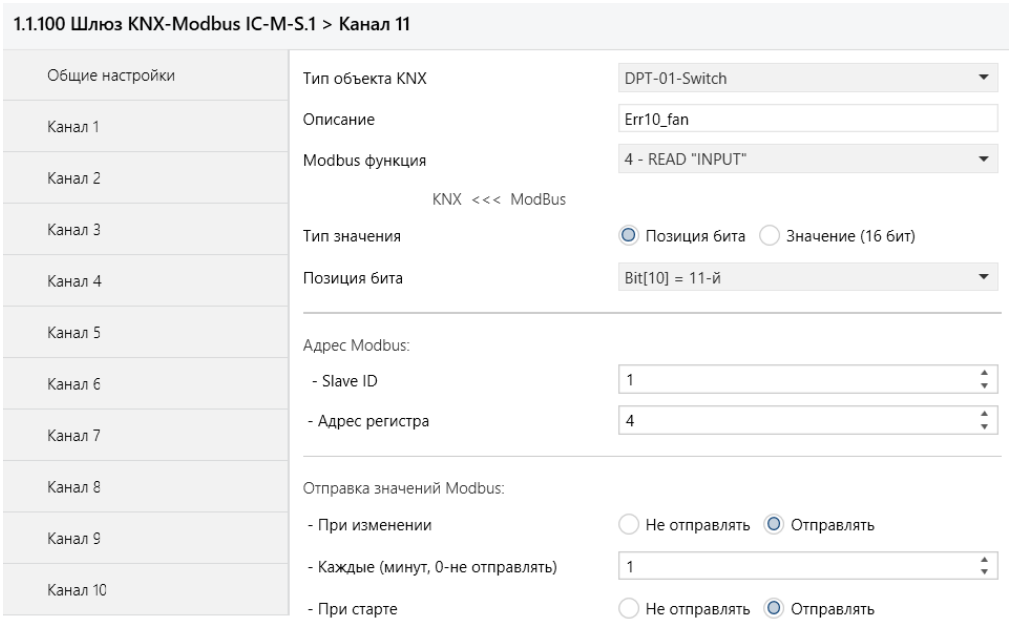
Коды ошибок и неисправностей ПВУ хранятся в регистре ввода #04 в битовом представлении (каждой ошибке соответствует свой бит) беззнакового целого двухбайтового числа. (см. раздел *Точки управления Modbus*). Так, неисправность датчика температуры T1 в канале 9 соответствует первому биту регистра #04.

10 Датчик давления вентилятора - Ошибка



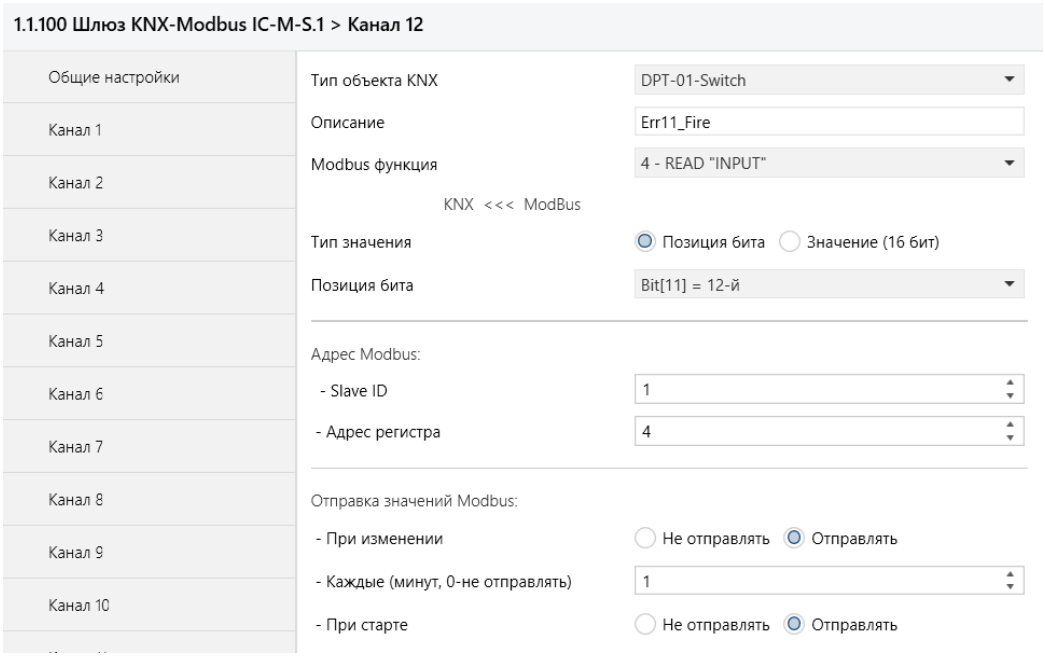
Настройка канала аналогична настройке канала 9, но с выделением десятого бита регистра #04.

11 Вентилятор приточный - Авария



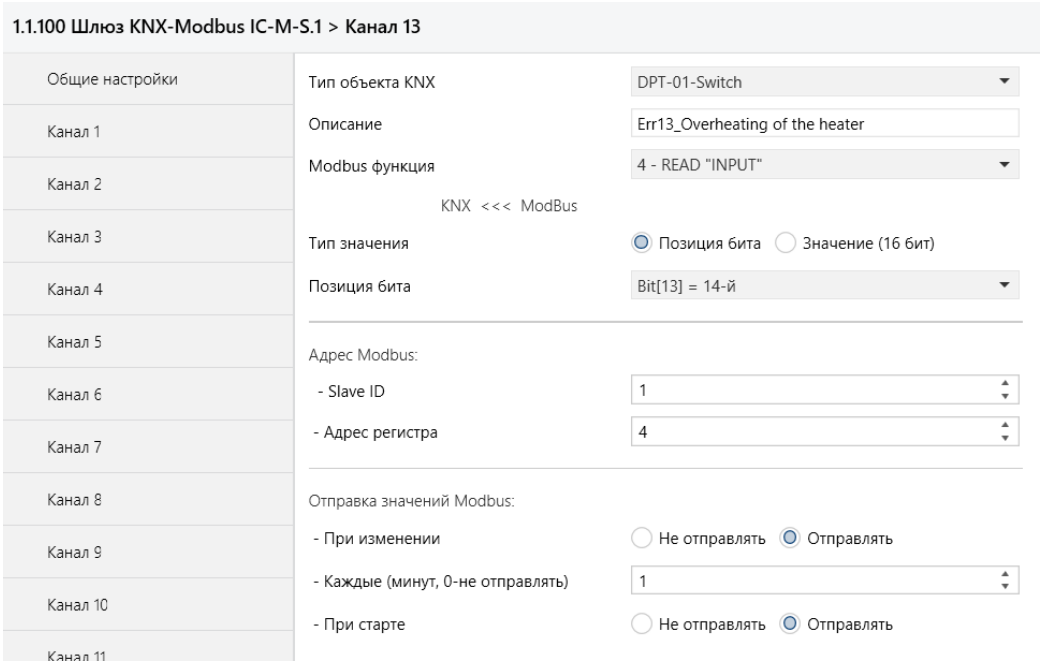
Настройка канала аналогична настройке канала 9, но с выделением одиннадцатого бита регистра #04.

12 Пожар



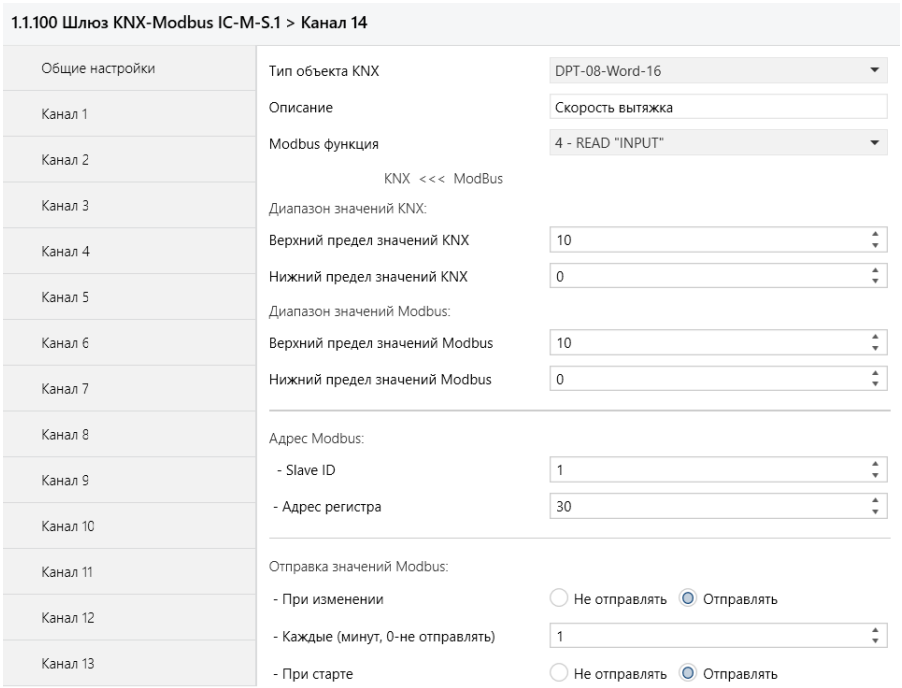
Настройка канала аналогична настройке канала 9, но с выделением двенадцатого бита регистра #04.

13 Перегрев



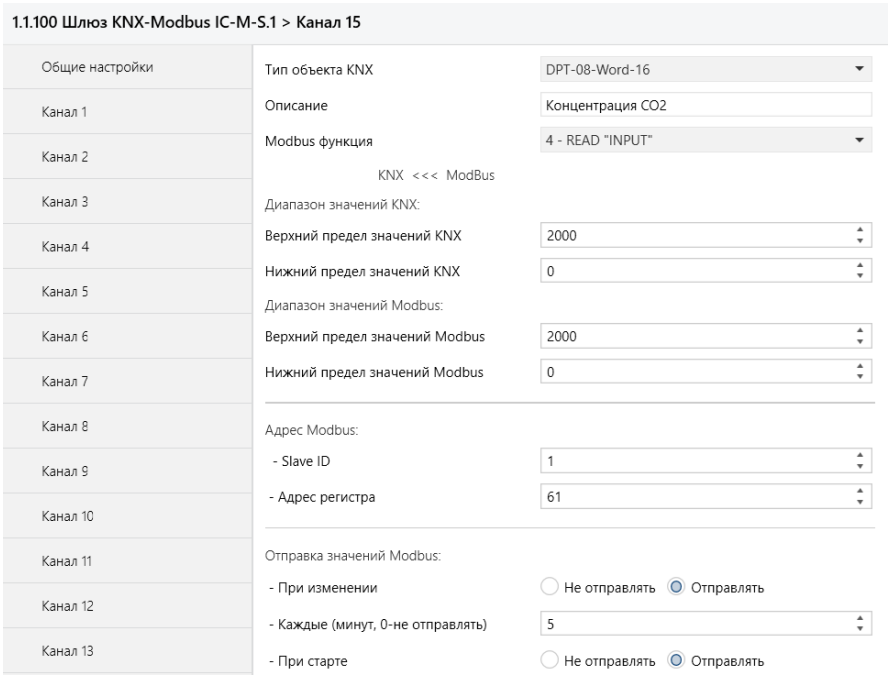
Настройка канала аналогична настройке канала 9, но с выделением четырнадцатого бита регистра #04.

14 Скорость вытяжного вентилятора



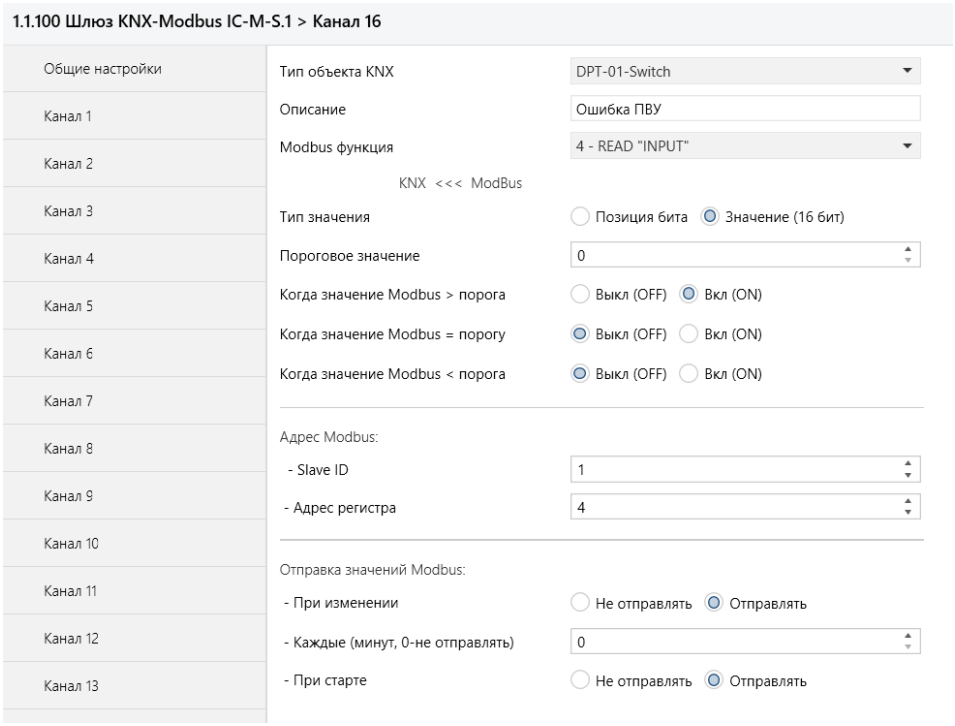
В ПВУ Ventmachine скорость вытяжного вентилятора задается целым беззнаковым числом в диапазоне от 1 до 10 (в режиме с аналоговым 0-10В управлением вентилятором), хранящемуся в регистре ввода #30. Значение 0 соответствует выключенному состоянию. В проекте используется синхронный режим регулирования приточного и вытяжного вентиляторов.

15 Концентрация CO_2



В соответствии с документацией ПВУ Ventmachine скорректированное значение концентрации CO_2 хранится в регистре ввода #61 в формате беззнакового целого числа в ppm. Предполагается, что измерения производятся внешим датчиком с выходом 0-10В и верхним пределом измерений 2000ppm.

16 Ошибка ПВУ – общий сигнал



В канале для стороны KNX формируется общий однобитовый сигнал ошибки ПВУ типа DPT-01-Switch, позволяющий вывести на пользовательский интерфейс KNX сообщение о неисправности ПВУ без ее детализации. Для этого шлюз выполняет преобразование двухбайтового беззнакового целого числа в однобитовое значение относительно порогового значения 0.

KNX проект.

После параметризации 16 каналов шлюза IC-M-S.1 в ETS у него становятся доступными следующие групповые объекты:

Групповые объекты 16 | Параметры

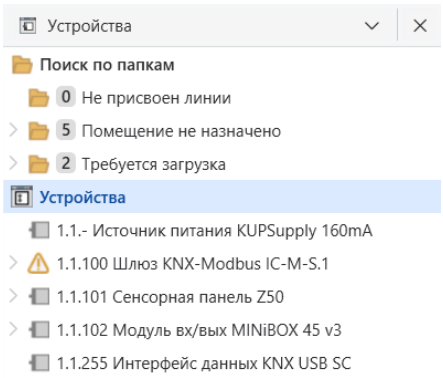
Номер	Имя	Объектная функция	Длина	Тип данных	C	R	W	T	U
1	Канал 01: ПВУ Вкл/Выкл	Write 1-bit	1 bit	switch	C	-	W	-	U
482	Канал 02: Скорость вент.	Write value-16	2 bytes	pulses difference	C	-	W	-	U
543	Канал 03: Скорость вент. Статус	Read value-16	2 bytes	pulses	C	R	-	T	-
64	Канал 04: ПВУ Вкл/Выкл Статус	Read 1-bit	1 bit	switch	C	R	-	T	-
605	Канал 05: Уставка T°C	Write float	2 bytes	temperature (°C)	C	-	W	-	U
666	Канал 06: Текущая температура	Read float	2 bytes	temperature (°C)	C	R	-	T	-
667	Канал 07: Текущая отн. влажность	Read float	2 bytes	temperature (°C)	C	R	-	T	-
668	Канал 08: Уставка T°C Статус	Read float	2 bytes	temperature (°C)	C	R	-	T	-
69	Канал 09: Err0_Датчик темп. ДТ1	Read 1-bit	1 bit	switch	C	R	-	T	-
70	Канал 10: Err9_Pressure sensor of fan	Read 1-bit	1 bit	switch	C	R	-	T	-
71	Канал 11: Err10_fan	Read 1-bit	1 bit	switch	C	R	-	T	-
72	Канал 12: Err11_Fire	Read 1-bit	1 bit	switch	C	R	-	T	-
73	Канал 13: Err13_Overheating of the heater	Read 1-bit	1 bit	switch	C	R	-	T	-
554	Канал 14: Скорость вытяжка	Read value-16	2 bytes	pulses difference	C	R	-	T	-
555	Канал 15: Концентрация CO2	Read value-16	2 bytes	pulses difference	C	R	-	T	-
76	Канал 16: Ошибка ПВУ	Read 1-bit	1 bit	switch	C	R	-	T	-

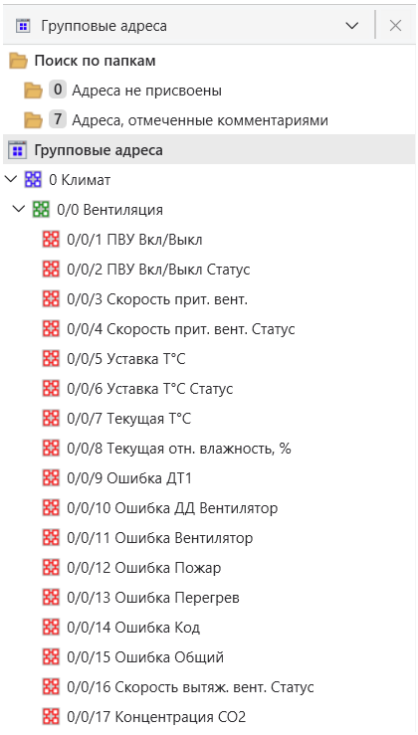
В свою очередь параметризация управляющих элементов сенсорной панели Z50 активирует следующий перечень её групповых объектов:

Каналы 2 | Групповые объекты 30 | Параметры | Поис

Номер	Имя	Объектная функция	Длина	Тип данных	C	R	W	T	U
601	[C1][Вкл/Выкл] Переключатель - Управление	1-битное общее управ...	1 bit	switch	C	R	-	T	-
606	[C2][Скор. прит. вент.] Целое число - 2-байтовый беззнаковый статус	0 ... 65535	2 bytes	pulses	C	-	W	T	U
607	[C2][Скор. прит. вент.] Целое число - 2-байтовое беззнаковое управление	0 ... 65535	2 bytes	pulses	C	R	-	T	-
612	[C3][Уставка T°C] Уставка - Температурный статус	-99°C ... 199°C	2 bytes	temperature (°C)	C	-	W	T	U
613	[C3][Уставка T°C] Уставка - Управление температурой	-99°C ... 199°C	2 bytes	temperature (°C)	C	R	-	T	-
615	[C3][Уставка T°C] Уставка - статус температуры окружающей среды	-99°C ... 199°C	2 bytes	temperature (°C)	C	-	W	T	U
618	[C4][Отн. влажность, %] Плав. - 2-байтовый статус	-671088.64 ... 670433.28	2 bytes	humidity (%)	C	-	W	T	U
624	[C5][Неисправность ПВУ] Переключатель - Статус	1-битный индикатор	1 bit	switch	C	-	W	T	U
630	[C6][Скор. вытяж. вент.] Целое число - 2-байтовый беззнаковый статус	0 ... 65535	2 bytes	pulses	C	-	W	T	U
636	[C7][Концентрация CO2] Целое число - 2-байтовый беззнаковый статус	0 ... 65535	2 bytes	pulses	C	-	W	T	U
642	[C8][Датчик T1] Переключатель - Статус	1-битный индикатор	1 bit	switch	C	-	W	T	U
648	[C9][Датчик давл. вент.] Переключатель - Статус	1-битный индикатор	1 bit	switch	C	-	W	T	U
654	[C10][Вент. прит. авария] Переключатель - Статус	1-битный индикатор	1 bit	switch	C	-	W	T	U
660	[C11][Пожар] Переключатель - Статус	1-битный индикатор	1 bit	switch	C	-	W	T	U
666	[C12][Перегрев калорифер] Переключатель - Статус	1-битный индикатор	1 bit	switch	C	-	W	T	U
672	[C13][Ошибки ПВУ] Перечень устройств - Статус	0 ... 255	1 byte	counter pulses (0..25...	C	-	W	T	U

Находящиеся ниже скриншоты из ETS иллюстрируют перечень использованных в проекте KNX устройств, созданные групповые адреса и групповые связи:





Объект	Устройство	Длина	Тип данных	C	R	W	T	U
0/0/1 ПВУ Вкл/Выкл								
1: Канал 01: ПВУ Вкл/Выкл - Write 1-bit	1.1.100 Шлюз KNX-Modbus IC-M-S.1	1 bit	switch	C	-	W	-	U
601: [C1][Вкл/Выкл] Переключатель - Управление - 1-битное общее управление	1.1.101 Сенсорная панель Z50	1 bit	switch	C	R	-	T	-
0/0/2 ПВУ Вкл/Выкл Статус								
64: Канал 04: ПВУ Вкл/Выкл Статус - Read 1-bit	1.1.100 Шлюз KNX-Modbus IC-M-S.1	1 bit	switch	C	R	-	T	-
600: [C1][Вкл/Выкл] Переключатель - Статус - 1-битный индикатор	1.1.101 Сенсорная панель Z50	1 bit	switch	C	-	W	T	U
0/0/3 Скорость прит. вент.								
482: Канал 02: Скорость вент. - Write value-16	1.1.100 Шлюз KNX-Modbus IC-M-S.1	2 bytes	pulses difference	C	-	W	-	U
607: [C2][Скор. прит. вент.] Целое число - 2-байтовое беззнаковое управление...	1.1.101 Сенсорная панель Z50	2 bytes	pulses	C	R	-	T	-
0/0/4 Скорость прит. вент. Статус								
543: Канал 03: Скорость вент. Статус - Read value-16	1.1.100 Шлюз KNX-Modbus IC-M-S.1	2 bytes	pulses	C	R	-	T	-
606: [C2][Скор. прит. вент.] Целое число - 2-байтовый беззнаковый статус - 0	1.1.101 Сенсорная панель Z50	2 bytes	pulses	C	-	W	T	U
0/0/5 Уставка T°C								
605: Канал 05: Уставка T°C - Write float	1.1.100 Шлюз KNX-Modbus IC-M-S.1	2 bytes	temperature (°C)	C	-	W	-	U
613: [C3][Уставка T°C] Уставка - Управление температурой - -99°C ... 199°C	1.1.101 Сенсорная панель Z50	2 bytes	temperature (°C)	C	R	-	T	-
0/0/6 Уставка T°C Статус								
612: [C3][Уставка T°C] Уставка - Температурный статус - -99°C ... 199°C	1.1.101 Сенсорная панель Z50	2 bytes	temperature (°C)	C	-	W	T	U
668: Канал 08: Уставка T°C Статус - Read float	1.1.100 Шлюз KNX-Modbus IC-M-S.1	2 bytes	temperature (°C)	C	R	-	T	-
0/0/7 Текущая T°C								
615: [C3][Уставка T°C] Уставка - статус температуры окружающей среды - -99°C...	1.1.101 Сенсорная панель Z50	2 bytes	temperature (°C)	C	-	W	T	U
666: Канал 06: Текущая температура - Read float	1.1.100 Шлюз KNX-Modbus IC-M-S.1	2 bytes	temperature (°C)	C	R	-	T	-
0/0/8 Текущая отн. влажность, %								
618: [C4][Отн. влажность, %] Плав. - 2-байтовый статус - -671088.64 ... 670433.28	1.1.101 Сенсорная панель Z50	2 bytes	humidity (%)	C	-	W	T	U
667: Канал 07: Текущая отн. влажность - Read float	1.1.100 Шлюз KNX-Modbus IC-M-S.1	2 bytes	temperature (°C)	C	R	-	T	-
0/0/9 Ошибка ДТ1								
69: Канал 09: Err0_Датчик темп. ДТ1 - Read 1-bit	1.1.100 Шлюз KNX-Modbus IC-M-S.1	1 bit	switch	C	R	-	T	-
604: [ЛФ] (1 бит) Вход данных 1 - Вход двоичных данных (0/1)	1.1.102 Модуль вх/вых MINiBOX 45 v3	1 bit	boolean	C	-	W	-	-
642: [C8][Датчик T1] Переключатель - Статус - 1-битный индикатор	1.1.101 Сенсорная панель Z50	1 bit	switch	C	-	W	T	U
0/0/10 Ошибка ДД Вентилятор								
70: Канал 10: Err9_Pressure sensor of fan - Read 1-bit	1.1.100 Шлюз KNX-Modbus IC-M-S.1	1 bit	switch	C	R	-	T	-
605: [ЛФ] (1 бит) Вход данных 2 - Вход двоичных данных (0/1)	1.1.102 Модуль вх/вых MINiBOX 45 v3	1 bit	boolean	C	-	W	-	-
648: [C9][Датчик давл. вент.] Переключатель - Статус - 1-битный индикатор	1.1.101 Сенсорная панель Z50	1 bit	switch	C	-	W	T	U
0/0/11 Ошибка Вентилятор								
71: Канал 11: Err10_fan - Read 1-bit	1.1.100 Шлюз KNX-Modbus IC-M-S.1	1 bit	switch	C	R	-	T	-
606: [ЛФ] (1 бит) Вход данных 3 - Вход двоичных данных (0/1)	1.1.102 Модуль вх/вых MINiBOX 45 v3	1 bit	boolean	C	-	W	-	-
654: [C10][Вент. прит. авария] Переключатель - Статус - 1-битный индикатор	1.1.101 Сенсорная панель Z50	1 bit	switch	C	-	W	T	U
0/0/12 Ошибка Пожар								
72: Канал 12: Err11_Fire - Read 1-bit	1.1.100 Шлюз KNX-Modbus IC-M-S.1	1 bit	switch	C	R	-	T	-
607: [ЛФ] (1 бит) Вход данных 4 - Вход двоичных данных (0/1)	1.1.102 Модуль вх/вых MINiBOX 45 v3	1 bit	boolean	C	-	W	-	-
660: [C11][Пожар] Переключатель - Статус - 1-битный индикатор	1.1.101 Сенсорная панель Z50	1 bit	switch	C	-	W	T	U
0/0/13 Ошибка Перегрев								
73: Канал 13: Err13_Overheating of the heater - Read 1-bit	1.1.100 Шлюз KNX-Modbus IC-M-S.1	1 bit	switch	C	R	-	T	-
608: [ЛФ] (1 бит) Вход данных 5 - Вход двоичных данных (0/1)	1.1.102 Модуль вх/вых MINiBOX 45 v3	1 bit	boolean	C	-	W	-	-
666: [C12][Перегрев калорифер] Переключатель - Статус - 1-битный индикатор	1.1.101 Сенсорная панель Z50	1 bit	switch	C	-	W	T	U
0/0/14 Ошибка Код								
672: [C13][Ошибки ПВУ] Перечень устройств - Статус - 0 ... 255	1.1.101 Сенсорная панель Z50	1 byte	counter pulses (0..255)	C	-	W	T	U
676: [ЛФ] Функция 1 - результат - (1 байт) Без знака	1.1.102 Модуль вх/вых MINiBOX 45 v3	1 byte	counter pulses (0..255)	C	R	-	T	-
677: [ЛФ] Функция 2 - результат - (1 байт) Без знака	1.1.102 Модуль вх/вых MINiBOX 45 v3	1 byte	counter pulses (0..255)	C	R	-	T	-
678: [ЛФ] Функция 3 - результат - (1 байт) Без знака	1.1.102 Модуль вх/вых MINiBOX 45 v3	1 byte	counter pulses (0..255)	C	R	-	T	-
679: [ЛФ] Функция 4 - результат - (1 байт) Без знака	1.1.102 Модуль вх/вых MINiBOX 45 v3	1 byte	counter pulses (0..255)	C	R	-	T	-
680: [ЛФ] Функция 5 - результат - (1 байт) Без знака	1.1.102 Модуль вх/вых MINiBOX 45 v3	1 byte	counter pulses (0..255)	C	R	-	T	-
0/0/15 Ошибка Общий								
76: Канал 16: Ошибка ПВУ - Read 1-bit	1.1.100 Шлюз KNX-Modbus IC-M-S.1	1 bit	switch	C	R	-	T	-
624: [C5][Неисправность ПВУ] Переключатель - Статус - 1-битный индикатор	1.1.101 Сенсорная панель Z50	1 bit	switch	C	-	W	T	U
0/0/16 Скорость вытяж. вент. Статус								
554: Канал 14: Скорость вытяжка - Read value-16	1.1.100 Шлюз KNX-Modbus IC-M-S.1	2 bytes	pulses difference	C	R	-	T	-
630: [C6][Скор. вытяж. вент.] Целое число - 2-байтовый беззнаковый статус - 0...	1.1.101 Сенсорная панель Z50	2 bytes	pulses	C	-	W	T	U
0/0/17 Концентрация CO2								
555: Канал 15: Концентрация CO2 - Read value-16	1.1.100 Шлюз KNX-Modbus IC-M-S.1	2 bytes	pulses difference	C	R	-	T	-
636: [C7][Концентрация CO2] Целое число - 2-байтовый беззнаковый статус - 0...	1.1.101 Сенсорная панель Z50	2 bytes	pulses	C	-	W	T	U

Проект инсталляции создавался в среде ETS 6.3.1 и доступен для скачивания по ссылке:

[Проект ETS KNX_ModBus_Ventmachine]

В качестве элемента тестирования инсталляции полезно записать и провести анализ межсистемного трафика с помощью любых доступных инструментальных средств мониторинга последовательных шин RS-485/RS-232/USB.

На рисунке ниже приведен фрагмент лога Modbus сообщений, записанных во время функционирования описываемой в статье инсталляции. На скриншоте пары сообщений «запрос-ответ» перенумерованы красным цветом.

Анализ принимаемого COM-портом потока данных

Файл Вид Сервис Справка

	Время	Данные	Комментарии
1.	19:15:42.333	01 04 00 3D 00 01 A0 06	Запрос регистров 3x0062-3x0062
	19:15:42.333	01 04 02 01 58 B9 5A	Ответ на запрос регистров - 2б данных(1рег)
2.	19:15:42.364	01 04 00 04 00 01 70 0B	Запрос регистров 3x0005-3x0005
	19:15:42.364	01 04 02 00 00 B9 30	Ответ на запрос регистров - 2б данных(1рег)
3.	19:15:42.380	01 06 00 02 00 01 E9 CA	Регистр 4x0003 установить в 1
	19:15:42.380	01 06 00 02 00 01 E9 CA	Регистр 4x0003 установлен в 1
4.	19:15:42.411	01 03 00 02 00 01 25 CA	Запрос регистров 4x0003-4x0003
	19:15:42.411	01 03 02 00 00 B8 44	Ответ на запрос регистров - 2б данных(1рег)
5.	19:15:42.426	01 03 00 20 00 01 85 C0	Запрос регистров 4x0033-4x0033
	19:15:42.426	01 03 02 00 05 78 47	Ответ на запрос регистров - 2б данных(1рег)
6.	19:15:42.458	01 04 00 02 00 01 90 0A	Запрос регистров 3x0003-3x0003
	19:15:42.458	01 04 02 01 43 F9 51	Ответ на запрос регистров - 2б данных(1рег)
7.	19:15:42.473	01 04 00 09 00 01 E1 C8	Запрос регистров 3x0010-3x0010
	19:15:42.473	01 04 02 00 EA 38 BF	Ответ на запрос регистров - 2б данных(1рег)
8.	19:15:42.505	01 04 00 3A 00 01 11 C7	Запрос регистров 3x0059-3x0059
	19:15:42.505	01 04 02 00 2C B8 ED	Ответ на запрос регистров - 2б данных(1рег)
9.	19:15:42.520	01 03 00 1F 00 01 B5 CC	Запрос регистров 4x0032-4x0032
	19:15:42.520	01 03 02 00 ED 78 09	Ответ на запрос регистров - 2б данных(1рег)
10.	19:15:42.551	01 04 00 04 00 01 70 0B	Запрос регистров 3x0005-3x0005
	19:15:42.551	01 04 02 00 00 B9 30	Ответ на запрос регистров - 2б данных(1рег)
11.	19:15:42.567	01 04 00 04 00 01 70 0B	Запрос регистров 3x0005-3x0005
	19:15:42.567	01 04 02 00 00 B9 30	Ответ на запрос регистров - 2б данных(1рег)
12.	19:15:42.598	01 04 00 04 00 01 70 0B	Запрос регистров 3x0005-3x0005
	19:15:42.598	01 04 02 00 00 B9 30	Ответ на запрос регистров - 2б данных(1рег)
13.	19:15:42.614	01 04 00 04 00 01 70 0B	Запрос регистров 3x0005-3x0005
	19:15:42.614	01 04 02 00 00 B9 30	Ответ на запрос регистров - 2б данных(1рег)
14.	19:15:42.645	01 04 00 04 00 01 70 0B	Запрос регистров 3x0005-3x0005
	19:15:42.645	01 04 02 00 00 B9 30	Ответ на запрос регистров - 2б данных(1рег)
15.	19:15:42.661	01 04 00 1E 00 01 51 CC	Запрос регистров 3x0031-3x0031
	19:15:42.661	01 04 02 00 00 B9 30	Ответ на запрос регистров - 2б данных(1рег)

В качестве полезной практики ниже приводятся расшифровки нескольких пар сообщений запрос-ответ:

1. Запрос к ведомому устройству #1 (**01**) с помощью функции чтения регистров ввода (**04**), начиная с адреса 30062 (**00 3D**) в количестве одного регистра (**00 01**) и полем контроля четности CRC (**A0 06**).

   Ответ мастеру от ведомого устройства #1 на запрос чтения регистров ввода (**01 04**), с полем данных общей длиной 2 байта (**02**), содержанием запрашиваемого регистра (**01 58**) и полем контроля четности CRC (**B9 5A**).

   В данном регистре хранится значение концентрации CO_2 в формате беззнакового целого в ppm. Пересчет шестнадцатричного значения (01 58) в десятичный формат дает значение 314 ppm.

2. Запрос к ведомому устройству #1 (**01**) с помощью функции чтения регистров ввода (**04**), начиная с адреса 30005 (**00 04**) в количестве одного регистра (**00 01**) и полем контроля четности CRC (**70 0B**).

Ответ мастеру от ведомого устройства #1 на запрос чтения регистров ввода (**01 04**), с полем данных общей длиной 2 байта (**02**), содержанием запрашиваемого регистра (**00 00**) и полем контроля четности CRC (**B9 30**).

В данном регистре хранятся в битовом представлении ошибки ПВУ в формате беззнакового целого. Нулевое значение содержимого регистра говорит об отсутствии ошибок.

3. Запрос к ведому устройству #1 (**01**) с помощью функции записи регистров хранения (**06**) на запись в регистр с адресом 40003 (**00 02**) значения 1 (**00 01**) и полем контроля четности CRC (**E9 CA**). Через младший бит регистра 40003 производится включение и выключение ПВУ.

   Ответ мастеру от ведомого устройства #1 повторяет запрос. Это означает, что команда мастера ведомым устройством принята и выполнена.

4. *Произведите анализ сообщений самостоятельно.*

5. Запрос к ведомому устройства #1 (**01**) с помощью функции чтения регистров хранения (**03**), начиная с адреса 40033 (**00 20**) в количестве одного регистра (**00 01**) и полем контроля четности CRC (**85 C0**).

   Ответ мастеру от ведомого устройства #1 на запрос чтения регистров хранения (**01 03**), с полем данных общей длиной 2 байта (**02**), содержанием запрашиваемого регистра (**00 05**) и полем контроля четности CRC (**78 47**).

   В данном регистре хранится текущее значение скорости приточного вентилятора в формате беззнакового целого. Шестнадцатричное число (**00 05**) означает 5-ю скорость.

6. Запрос к ведомому устройства #1 (**01**) с помощью функции чтения регистров ввода (**04**), начиная с адреса 30003 (**00 02**) в количестве одного регистра (**00 01**) и полем контроля четности CRC (**90 0A**).

   Ответ мастеру от ведомого устройства #1 на запрос чтения регистров ввода (**01 04**), с полем данных общей длиной 2 байта (**02**), содержанием запрашиваемого регистра (**01 43**) и полем контроля четности CRC (**F9 51**).

   В регистре 30003 хранятся в битовом представлении данные о режиме работы ПВУ в формате беззнакового целого. В частности, младший бит данного регистра содержит статус ПВУ включено (1)-выключено (0). В данном случае ПВУ находится в состоянии «включено».

7. Запрос к ведомому устройству #1 (**01**) с помощью функции чтения регистров ввода (**04**), начиная с адреса 30010 (**00 09**) в количестве одного регистра (**00 01**) и полем контроля четности CRC (**E1 C8**).

   Ответ мастеру от ведомого устройства #1 на запрос чтения регистров ввода (**01 04**), с полем данных общей длиной 2 байта (**02**), содержанием запрашиваемого регистра (**00 EA**) и полем контроля четности CRC (**38 BF**).

   В регистре 30010 хранится текущая температура T1 в канале ПВУ в формате знакового целого в десятых долях °C. В данном случае текущая температура составляет:

   (14x16+10)x0.1=23.4 °C.

8-15. *Произведите анализ сообщений самостоятельно.*

Приложения.

Ссылки на дополнительные материалы к статье:

1. Шлюз Modbus-KNX IC-M-S.1

   Аппликационная программа | Руководство пользователя

2. Сенсорная панель Zennio Z50

   Аппликационная программа | Руководство пользователя

3. Спецификация регистров Modbus RTU ПВУ Ventmachine (GTC v5)
4. Проект ETS KNX_ModBus_Ventmachine
5. Видео Ventmachine_GTC_Z50_KNX (RuTube)

   Видео Ventmachine_GTC_Z50_KNX (YouTube)

6. Полезные интернет-ресурсы:
   https://www.modbus.org/specs.php
   https://en.wikipedia.org/wiki/Modbus#CITEREFMODBUS_Application_Protocol2012
   https://habr.com/ru/companies/advantech/articles/450234/

Тренинговый центр **Ассоциации KNX**
Laboratory I-Home.ru

Многоступенчатая программа обучения и
сертификации KNX специалистов.

Ждем Вас на наших курсах.

KNXdesk.com
trainings@knxdesk.com
г.Москва, ул.Халтуринская д.6А
8 800 511 8911 (бесплатный)
+7 495 988 7593 | +7 499 748 1173