

## Датчик с интерфейсом LoRa WAN V2.5

### Общее описание

Датчик - устройство в стандарте LoRa WAN класс A  
Конкретный набор функций зависит от исполнения.

Поддерживаемые сенсоры

- SI7021 - интерфейс I2c
- DS18B20 - интерфейс 1-Wire
- AM2302/DHT22 (Только при питании 3,3 –5 вольт)
- PT100 (микросхема MAX31865) - интерфейс SPI (Только при питании 3,3 –5 вольт)
- АЦП (Только при питании 3,3 –5 вольт)
- Счетчик импульсов 4 канала 32 бит, с возможностью выбора - геркон или НАМУР для каждого канала
- Работа этих входов датчика, как 4 входа (при этом счетчик отключен) – моментальная передача при срабатывании. Выбор канала индивидуально.
- Работа этих входов, как 4 выхода (при этом сенсоры отключены, счетчик отключен).
- часы реального времени + календарь.
- Датчик может принимать нисходящие сообщения и управлять реле.
- контроль источника питания (аккумулятора/батареи)

В датчике есть BOOT загрузчик для обновления встроенного ПО

Датчик имеет интерфейс UART TTL для конфигурации и обновления прошивки

Питание датчика - 2 батарейки AA Alkaline 2,0 -3,3 Вольт Вариант -1

Питание датчика - 3 батарейки AA Alkaline 3,3 – 5 Вольт, литиевый аккумулятор, внешний источник 5,0 вольт. Вариант -2 (В случае применения MAX31865, DHT22, АЦП)

Питание 9 вольт - вариант 3

### Общее описание включения

Включение питания

Микроконтроллер включен всегда, пока есть питание. Находится в режиме сна. В заданное время просыпается и выполняет нужные действия. При отсутствии питания сбрасываются часы и счетчики. Настройки хранятся в энергонезависимой памяти.

Кнопка RESET

Приводит к перезагрузке контроллера, запуске повторной активации в сети LoRa.

Включение питания или нажатие RESET

- Стартует загрузчик и ждет 3 секунды соединения с программой загрузчика.
- Если нет соединения, то передается управления основной программе при ее наличии. Если ее нет, то остается активен загрузчик.
- Стартует основная программа. Зажигаются оба светодиода. Ожидается соединение с программой управления и настройки в течении двух секунд. Если соединения не произошло, то запускается обычная работа основной программы. Иначе запускается режим настройки.
- Выход из режима настройки – команда в терминале, или таймаут 10 мин. Далее обычная работа
- Датчик опрашивает включенные сенсоры и передает их значения во время отправки. Могут быть настроены границы нормальных значений, переход этих границ генерирует события. Они могут быть отправлены во время обычного сеанса связи, либо можно настроить, что событие инициирует немедленную отправку.

### Энергопотребление

Потребление энергии сильно зависит от дальности до шлюза. Указаны типичные значения.

Режим сна, датчики обесточены. Радио модуль IDLE ~17(L) - 30 мкА

Передача на максимальной мощности ~ 90 мА в течении 1-2 сек. И прием 3-4 сек 20-25 мА

Минимальная мощность Переда + прием 7 мА 1 сек.

Потребление самого микроконтроллера PIC в активном режиме 4,5 мА (опрос датчиков)

Примерный расчет времени работы от батарей

Емкость 2 шт. батареек щелочных AA = 40 000 сеансов связи на максимальной мощности.

Сам датчик уменьшает мощность вплоть до минимальной при хорошем качестве связи и число передач может вырасти на порядок. При этом число сеансов связи может достигать 560 тысяч. Справка - в году 525600 минут

### Обмен данными, уровень шлюза

MQTT работает. При этом thethingsnetwork.org не нужен.

Доставляются и правильно декодируются нисходящие сообщения

Не теряются сообщения MQTT, если был отключен (включается в настройках)

## Настройка и подготовка к работе.

Модуль из коробки сразу можно включить и проверить. Он настроен согласно таблицы, значениями по умолчанию. Для нормальной работы следует выполнить настройки параметров.

### Настройка сети LoRa (требуется программа конфигуратор)

- Тип присоединения к LoRa сети = OTAA
- Device EUI (DEV EUI) – номер устройства
- Application EUI (APP EUI) - вариант восходящих данных, типа устройства с одинаковыми настройками
- Device Address (Dev ADDR) - это номер в конкретной сети? Как MAC
- Application session key, Network session key, Application Key – ключи шифрования
- Power MODE (тип питания) 0 – от 2-х батареек, 1 = литий, 3 батарейки или внешнее
- Часы и календарь в формате 12:59:25 1-02-24#1

### Настройка данных и событий для восходящего канала

- Интервал передачи восходящих данных, в секундах
  - Настройки времени и временных интервалов.
  - Настройка поведения по времени
    - Включить в содержание посылки значение времени
    - включить отправку посылки каждый час, в указанное значение минут
    - включить отправку посылки посылку раз в день, в указанный час
    - включить отправку посылки раз в месяц, в указанный день
    - включить отправку посылки раз в неделю, в указанный день
- добавить время тишины. Указывается интервал времени, когда запрещены посылки. Нисходящий канал тоже отключен. Все события в данном интервале сбрасываются и не обрабатываются. В период блокировки все события к отправке будут сброшены, интервалы обнулены, при их наступлении.

Включить режим принудительного пере соединения со шлюзом, перезагрузку и заново присоединение к сети LoRa через указанное количество часов. Шлюз может блокировать датчики при большой загрузке эфира и множестве восходящих каналов.

- настройка подключенных датчиков

В качестве датчиков выступают: DS18B20, si7021 (SHT21), AM2302/DHT22, 1 PT100 (микросхема MAX31865), 4 Канала АЦП.

Если какой-либо датчик включен, то его состояние передается при любом восходящем сообщении - настройка поведения при обработке датчика. Настройка индивидуальных нижних и верхних пороговых значений для каждого типа. Настройка поведения при переходе порогов. Событием является переход верхнего порога снизу-вверх, генерируется однократно. Для повторного требуется опускание значения ниже нижнего порога, будет генерация события нижнего порога. И при следующем превышении верхнего снова событие. Таким образом события - это выход из коридора между порогами. Следите за правильными значениями порогов.

Интервал опроса датчика в секундах.

Передача события во время обычного сеанса связи по расписанию.

Событие инициирует немедленную передачу, но не чаще времени опроса датчика

Привязка событий к датчикам и порогам индивидуально

- настройка датчика протечек

Включение/выключение

Интервал опроса

Настройка событий по замыканию и размыканию.

Немедленная или передача по расписанию

- настройка входов, как цифровых с поддержкой NAMUR (индивидуально)

настройка типа, цифровой или NAMUR

настройка поведения, счетчик или вход. Если счетчик, то показания передаются при каждом сеансе передачи

настройка поведения передачи входа, событий по изменению состояния, типа передачи событий

\* Если вход отключен и также не подключено к нему АЦП, то он становится выходом и может управляться нисходящим каналом

настройка времени опроса цифрового входа с поддержкой NAMUR, сек

настройка времени опроса счетного входа с поддержкой NAMUR. Кратно 50 миллисекунд

Выполняется независимо от работы передатчика, для счета импульсов

### Дополнительные настройки

Только при использовании MAX31865

Тип подключения датчика 2/3/4 провода.

Номинал термосопротивления PT100 / PT1000  
 Номинал резистора Rref 430\*/4300\*  
 Коррекция температуры для каждого датчика  
 Коррекция влажности для каждого датчика  
 Время пульса реле, сек (должно быть меньше интервала связи нисходящего канала)  
 Обнуление счетчиков (для каждого канала)

#### Проверки из конфигуратора

- Статус источника питания: Напряжение внутреннего источника питания - это десятичное число, представляющее уровень заряда батареи, от 0 до 255; 0 означает внешнее питание, 1 - низкий уровень, 254- высокий уровень и 255 означает, что конечное устройство не смогло измерить уровень заряда батареи.
- Текущая температура DS18B20 (с коррекцией)
- Текущая температура si7021 (SHT21). (с коррекцией)
- Текущая влажность si7021 (SHT21). (с коррекцией)
- Текущая температура AM2302/DHT22(с коррекцией)
- Текущая влажность AM2302/DHT22(с коррекцией)
- Текущая температура PT100 (микросхема MAX31865) (с коррекцией)
- Значение счетчика 1.
- Значение счетчика 2.
- Значение счетчика 3.
- Значение счетчика 4.
- Значение ADC1 (текущее)
- Значение ADC2 (текущее)
- Значение ADC3 (текущее)
- Значение ADC4 (текущее)

#### Сервис

Управление Реле #1  
 Управление Реле #2  
 Управление Реле #3  
 Управление Реле #4

#### Описание регистров параметров для настройки

- Изменения, согласно значению регистра применяются немедленно, после ответа на команду изменения или после перезагрузки – указано дополнительно.

Описание	тип доступа	значение по умолчанию	допустимые значения
Интервал передачи восходящих данных, в секундах Требуется перезагрузка	R/W	20	15 – 3 000 000
Тип присоединения к LoRa сети: JOIN Mode: 0 = OTAA; 1 = ABP. Требуется перезагрузка	R/W	0	0 -1
Device EUI Требуется перезагрузка	R/W	0004A30B0 01A9EF8	64 бит число в HEX записи
Application session key Требуется перезагрузка	R/W	CADC8A1F B050F456E 93EDDD16 27D9945	128 бит число в HEX записи
Network session key Требуется перезагрузка	R/W	DAF8760C CD390EC8 D11AF5115 BC9B698	128 бит число в HEX записи
Application Key Требуется перезагрузка	R/W	3C8F26273 9BFE3B7BC 0826991A D0504D	128 бит число в HEX записи
Application EUI Требуется перезагрузка	R/W	112233445 5667788	64 бит число в HEX записи
Device Address Требуется перезагрузка	R/W	89002483	32 бит число в HEX записи
Power MODE (тип питания ) Требуется перезагрузка 0 - 2,0-3,3 вольт; две батарейки AA	R/W	0	0-255

1 – 3.3 – 4,2 (4,5) вольт. Три батарейки АА, Литий. Внешнее 2 – 6F22 9 вольт.			
Статус источника питания: Напряжение внутреннего источника питания - это десятичное число, представляющее уровень заряда батареи, от 0 до 255; 0 означает внешнее питание, 1 - низкий уровень, 254 - высокий уровень и 255 означает, что конечное устройство не смогло измерить уровень заряда батареи.	R	*	0-255
Часы и календарь в формате 12:59:25 1-02-24#1 Час:мин:сек день-месяц-год#день недели(1-понедельник)	R/W	12:59:25 1-02-24#1	
Битовый регистр настроек по времени. Бит 0 - Включить в посылку значение времени Бит 1 – включить посылку каждый час, значение минут для отправки берем из следующего регистра 14. Бит 2 – включить посылку раз в день. Значение часа и минут берем из регистра 14 Бит 3 – включить посылку раз в месяц. Значение дня, часа и минут берем из регистра 14 Бит 4 – включить посылку раз в день недели (например, вторник). Бит 5 – Включить блокировку по времени из регистра 15 Бит 6 – Ватчдог. Включить принудительную перезагрузку и заново присоединение к сети LoRa через указанное количество часов  Значение дня недели, часа и минут берем из регистра 14 При отправлении по времени, счетчик интервала передачи – 2 регистр, обнуляется	R/W	0-255	
Часы и календарь в формате 11:25:22#1 час:мин:день#день недели для получения интервалов для отправки сообщений по расписанию	R/W		
Начальный час и конечный час для настройки времени блокировки сообщений формате: Час начала блокировки: До какого часа заблокировать Например, 12:13:xx - заблокировано с 12:00 до 13:00 22:05:xx заблокировано с 22:00 до 5 утра В период блокировки все события к отправке будут сброшены, интервалы обнулены, при их наступлении. Третье значение – таймер времени соединения заново к сети, если включено.			
Настройка датчика - тип датчика: Битовый регистр Бит 0 = 1 DS18B20 задействован. 0 – отключен Бит 1 = 1 si7021 (SHT21) задействован. 0 – отключен Бит 2 = 1 AM2302/DHT22 задействован. 0 – откл. SCL - к линии AM2302/DHT22 Бит 3 = 1 PT100 (микросхема MAX31865) задействован. 0 – отключен Бит 4 = 1 Канал A_IN_1 задействован как АЦП. 0 – отключен (выход) Бит 5 = 1 Канал A_IN_2 задействован как АЦП. 0 – отключен Бит 6 = 1 Канал A_IN_3 задействован как АЦП. 0 – отключен Бит 7 = 1 Канал A_IN_4 задействован как АЦП. 0 – отключен Для использования АЦП не устанавливать подтяжку. Измерение производится по отношению к напряжению источнику питания. Если включено, то значение будет передаваться.	R/W	0	0 - 255
Настройка времени опроса датчика из предыдущего регистра, сек Выполняется независимо от работы передатчика, для	R/W	60	15 - 65535

инициирования события.			
<p>Настройка параметра отправки событий</p> <p>Бит 0 = 1 - включено по температуре во время обычной связи</p> <p>Бит 1 = 1 - превышение верхнего порога по температуре</p> <p>Бит 2 = 1 - ниже нижнего порога по температуре</p> <p>Бит 3 = 1 - Переход порога инициирует немедленную передачу, но не чаще времени опроса датчика. 0 – будет зафиксировано в следующей отправке данных по расписанию, если включено (Бит 0)</p> <p>Бит 4 = 1 - включено по влажности во время обычной связи</p> <p>Бит 5 = 1 - превышение верхнего порога по влажности</p> <p>Бит 6 = 1 - ниже нижнего порога по влажности</p> <p>Бит 7 = 1 - Переход порога инициирует немедленную передачу, но не чаще времени опроса датчика. 0 – будет зафиксировано в следующей отправке данных по расписанию, если включено (Бит 4)</p>	R/W		0 - 255
<p>Верхний порог температуры, записывается как 16 бит знаковый int16</p> <p>Температура в градусах Цельсия *10; например, +28,5 = 0x011D</p>	R/W	250	0-65535
Нижний порог температуры	R/W	100	0-65535
Верхний порог влажности, в процентах *10; например, 33% =330	R/W	500	0-65535
Нижний порог влажности	R/W	200	0-65535
<p>Настройка датчика протечки (цифровой вход) битовый регистр</p> <p>Бит 0 = 1 - D_IN_1 включен ;0- выключен</p> <p>Бит 1 = 1 - событие при замыкании включено</p> <p>Бит 2 = 1 - событие при размыкании включено</p> <p>Бит 3 = 1 - Переход порога инициирует немедленную передачу, но не чаще времени опроса датчика. 0 – будет зафиксировано в следующей отправке данных по расписанию</p> <p>Настройка датчика протечки (цифровой вход) битовый регистр</p> <p>Бит 4 = 1 - D_IN_2 включен ;0- выключен</p> <p>Бит 5 = 1 - событие при замыкании включено</p> <p>Бит 6 = 1 - событие при размыкании включено</p> <p>Бит 7 = 1 - Переход порога инициирует немедленную передачу, но не чаще времени опроса датчика. 0 – будет зафиксировано в следующей отправке данных по расписанию</p>	R/W		0 - 255
<p>Настройка времени опроса датчика из 23 регистра, сек</p> <p>Выполняется независимо от работы передатчика, для инициирования события.</p>	R/W	60	15 - 65535
<p>Настройка событий канала АЦП –</p> <p>A_IN_1</p> <p>Бит 0 = 1 - включено во время обычной связи</p> <p>Бит 1 = 1 - событие по превышению включено</p> <p>Бит 2 = 1 - событие по нижнему порогу включено</p> <p>Бит 3 = 1 - Переход порога инициирует немедленную передачу, но не чаще времени опроса датчика. 0 – будет зафиксировано в следующей отправке данных по расписанию, если включено</p> <p>A_IN_2</p> <p>Бит 4 = 1 - включено во время обычной связи</p> <p>Бит 5 = 1 - событие по превышению включено</p> <p>Бит 6 = 1 - событие по нижнему порогу включено</p> <p>Бит 7 = 1 - Переход порога инициирует немедленную передачу, но не чаще времени опроса датчика. 0 – будет зафиксировано в следующей отправке данных по расписанию, если включено</p>	R/W		0 - 255

<p>Настройка событий канала АЦП – A_IN_3 Бит 0 = 1 - включено во время обычной связи Бит 1 = 1 – событие по превышению включено Бит 2 = 1 – событие по нижнему порогу включено Бит 3 = 1 - Переход порога инициирует немедленную передачу, но не чаще времени опроса датчика. 0 – будет зафиксировано в следующей отправке данных по расписанию, если включено</p> <p>A_IN_4 Бит 4 = 1 - включено во время обычной связи Бит 5 = 1 – событие по превышению включено Бит 6 = 1 – событие по нижнему порогу включено Бит 7 = 1 - Переход порога инициирует немедленную передачу, но не чаще времени опроса датчика. 0 – будет зафиксировано в следующей отправке данных по расписанию, если включено</p>	R/W		0 - 255
Верхний порог АЦП1	R/W		0-1023
Нижний порог АЦП1	R/W		0-1023
Верхний порог АЦП2	R/W		0-1023
Нижний порог АЦП2	R/W		0-1023
Верхний порог АЦП3	R/W		0-1023
Нижний порог АЦП3	R/W		0-1023
Верхний порог АЦП4	R/W		0-1023
Нижний порог АЦП4	R/W		0-1023
<p>Настройка входа AIN - битовый регистр</p> <p>Бит 0 = 1 Канал A_IN_1 задействован как счетный вход 0 – отключен. Бит 1 = 1 Канал A_IN_1 – тип входа NAMUR; 0 – обычный Бит 2 = 1 Событие по высокому уровню на входе Бит 3 = 1 Событие по низкому уровню на входе Бит 4 = 1 Включать передачу значения счетчика Бит 5 = 1 Включать передачу значения цифрового входа Бит 6 = 1 Включать передачу по событию Бит 7 = 1 Канал A_IN_1 задействован как цифровой вход; 0 – отключен.</p> <p>Если вход отключен и также не подключено к нему АЦП, то он становится выходом и может управляться нисходящим каналом Каждый счетчик - 64 бит Значение счетчика сбрасывается при пропадании питания. Предаётся при каждом сеансе по расписанию, если включено. Требуется первичной инициализации</p>		0 - 255	0 - 255
<p>Настройка входа AIN - битовый регистр</p> <p>Бит 0 = 1 Канал A_IN_2 задействован как счетный вход 0 – отключен. Бит 1 = 1 Канал A_IN_2 – тип входа NAMUR; 0 – обычный Бит 2 = 1 Событие по высокому уровню на входе Бит 3 = 1 Событие по низкому уровню на входе Бит 4 = 1 Включать передачу значения счетчика Бит 5 = 1 Включать передачу значения цифрового входа Бит 6 = 1 Включать передачу по событию Бит 7 = 1 Канал A_IN_2 задействован как цифровой вход 0 – отключен.</p> <p>Если вход отключен и также не подключено к нему АЦП, то он становится выходом и может управляться нисходящим каналом Каждый счетчик - 64 бит Значение счетчика сбрасывается при пропадании питания. Предаётся при каждом сеансе по расписанию, если включено. Требуется первичной инициализации</p>			0 - 255

<p>Настройка входа AIN - битовый регистр</p> <p>Бит 0 = 1 Канал A_IN_3 задействован как счетный вход 0 – отключен.  Бит 1 = 1 Канал A_IN_3 – тип входа NAMUR; 0 – обычный  Бит 2 = 1 Событие по высокому уровню на входе  Бит 3 = 1 Событие по низкому уровню на входе  Бит 4 = 1 Включать передачу значения счетчика  Бит 5 = 1 Включать передачу значения цифрового входа  Бит 6 = 1 Включать передачу по событию  Бит 7 = 1 Канал A_IN_3 задействован как цифровой вход 0 – отключен.</p> <p>Если вход отключен и также не подключено к нему АЦП, то он становится выходом и может управляться нисходящим каналом  Каждый счетчик - 64 бит  Значение счетчика сбрасывается при пропадании питания.  Предаётся при каждом сеансе по расписанию, если включено. Требуется первичной инициализации по расписанию, если включено.</p>			0 - 255
<p>Настройка входа AIN - битовый регистр</p> <p>Бит 0 = 1 Канал A_IN_4 задействован как счетный вход 0 – отключен.  Бит 1 = 1 Канал A_IN_4 – тип входа NAMUR; 0 – обычный  Бит 2 = 1 Событие по высокому уровню на входе  Бит 3 = 1 Событие по низкому уровню на входе  Бит 4 = 1 Включать передачу значения счетчика  Бит 5 = 1 Включать передачу значения цифрового входа  Бит 6 = 1 Включать передачу по событию  Бит 7 = 1 Канал A_IN_4 задействован как цифровой вход 0 – отключен.</p> <p>Если вход отключен и также не подключено к нему АЦП, то он становится выходом и может управляться нисходящим каналом  Каждый счетчик - 64 бит  Значение счетчика сбрасывается при пропадании питания.  Предаётся при каждом сеансе по расписанию, если включено. Требуется первичной инициализации</p>			0 - 255
<p>Настройка времени опроса входа A_IN_x (цифровой вход) сек</p> <p>Выполняется независимо от работы передатчика, для инициализации события.</p>	R/W	60	1 - 65535
<p>Настройка времени опроса входа A_IN_x (счетный вход с поддержкой NAMUR). Кратно 50 миллисекунд</p> <p>Выполняется независимо от работы передатчика, для счета импульсов</p>	R/W	2	1-100
<p>Значение счетчика 1. Запись в регистр обнуляет счетчик</p>	R/W		
<p>Значение счетчика 2. Запись в регистр обнуляет счетчик</p>	R/W		
<p>Значение счетчика 3. Запись в регистр обнуляет счетчик</p>	R/W		
<p>Значение счетчика 4. Запись в регистр обнуляет счетчик</p>	R/W		
<p>Только при использовании MAX31865  Тип подключения датчика 2/3/4 провода.</p>	R/W	2/3/4	
<p>Только при использовании MAX31865  Номинал термосопротивления  PT100 / PT1000</p>	R/W	100	100 или 1000
<p>Только при использовании MAX31865  Номинал резистора Rref</p>	R/W	430	100 - 10000
<p>Коррекция температуры DS18B20. От - 12,8 до +12,7 град.  8 бит со знаком</p>	R/W	0	0-255
<p>Коррекция температуры si7021 (SHT21). От - 12,8 до +12,7 град.  8 бит со знаком</p>	R/W	0	0-255

Коррекция влажности si7021 (SHT21). От - 12,8% до 12,7%. 8 бит со знаком	R/W	0	0-255
Коррекция температуры AM2302/DHT22. От - 12,8 до 12,7 град. 8 бит со знаком.	R/W	0	0-255
Коррекция влажности AM2302/DHT22 от - 12,8% до 12,7%. 8 бит со знаком	R/W	0	0-255
Коррекция температуры PT100 (микросхема MAX31865). От - 12,8 до +12,7 град. 8 бит со знаком	R/W	0	0-255
Время пульса реле, сек (должно быть меньше интервала связи нисходящего канала)	R/W	1	1-255

### Примечание

Чем больше нагрузки на обработку датчиков и счетчиков, тем больше расход батарей питания. Чем выше частота опроса, тем больше расход батарей питания

### Восходящий канал. Формат данных. Принцип декодирования.

Все данные от датчиков. Значения параметров, и события включаются в восходящую посылку. Максимальный объём данных 51 байт. Лишние будут отсекаются. Младший байт идет первым. Первые пять байт передаются всегда. Их порядок не изменен. Остальные, согласно битам регистра статуса данных в этой посылке.

Формат посылки:

Байт #1 и #2 - Статус данных, подключенных к восходящему каналу. Описывает правило декодирования.

Бит #0 - Включена передача времени - 7 байт в формате 12:59:25 1-02-24#1 Час:мин:сек день-месяц-год# день недели(1- понедельник)

Бит #1 - Включена передача температуры датчика DS18B20 в формате int16 с фиксированной точкой 0,1 град С. Допустимые значения от -550 до +1250, что равно -55,0 + 125,0 град. Если датчик неисправен, то значение -32767.

Бит #2 - Включена передача температуры датчика SI7021 (SHT21) в формате int16 с фиксированной точкой 0,1 град С. Допустимые значения от -550 до +1250, что равно -55,0 + 125,0 град. Если датчик неисправен, то значение -32767.

Бит #3 - Включена передача влажности датчика SI7021 (SHT21) в формате int16 с фиксированной точкой 0,1 %. Допустимые значения от 0 до 999, что равно 0.0% - 99.9% Если датчик неисправен, то значение -32767.

Бит #4 - Включена передача температуры датчика AM2302/DHT22 в формате int16 с фиксированной точкой 0,1 град С. Допустимые значения от -550 до +1250, что равно -55,0 + 125,0 град. Если датчик неисправен, то значение -32767.

Бит #5 - Включена передача влажности датчика AM2302/DHT22 в формате int16 с фиксированной точкой 0,1 %. Допустимые значения от 0 до 999, что равно 0.0% - 99.9% Если датчик неисправен, то значение -32767.

Бит #6 - Включена передача температуры датчика PT100 в формате int16 с фиксированной точкой 0,1 град С. Допустимые значения от -2500 до +9000, что равно -250,0 + 9000,0 град. Если датчик неисправен, то значение -32767.

Бит #7 - Включена передача значения канала ADC#1 в формате int16. 0-1023

Байт №2 - старший:

Бит #8 - Включена передача значения канала ADC#2 в формате int16. 0-1023

Бит #9 - Включена передача значения канала ADC#3 в формате int16. 0-1023

Бит #10 - Включена передача значения канала ADC#4 в формате int16. 0-1023

Бит #11 - Включена передача значения канала DIN#1 - DIN#4, Включена передача значения канала датчика протечки и состояния реле - битовый регистр.

Бит #12 - Включена передача значения канала Счетчика #1 в формате uint32

Бит #13 - Включена передача значения канала Счетчика #2 в формате uint32

Бит #14 - Включена передача значения канала Счетчика #3 в формате uint32

Бит #15 - Включена передача значения канала Счетчика #4 в формате uint32

Байт #3 и #4 - Статус возникших событий, битовый регистр. Статус данных, подключенных к восходящему каналу. Описывает правило декодирования - продолжение.

Бит #0 - Событие превышения порога по температуре. Если включено два и более датчика, любой генерирует событие.



Бит #1 – Событие перехода вниз, нижнего порога по температуре. Если включено два и более датчика, любой генерирует событие

Бит #2 – Событие превышения порога по влажности. Если включено два датчика, любой генерирует событие.

Бит #3 – Событие перехода вниз, нижнего порога по влажности. Если включено два датчика, любой генерирует событие

Бит #4 – Событие наличия воды датчика протечки #1

Бит #5 – Событие наличия воды датчика протечки #2

Бит #6 – Событие отсутствия воды датчика протечки #1

Бит #7 – Событие отсутствия воды датчика протечки #2

Байт №4 - старший:

Бит #8 – Событие превышения верхнего порога ADC#1 или событие по высокому уровню на входе DIN#1

Бит #9 – Событие превышения верхнего порога ADC#2 или событие по высокому уровню на входе DIN#2

Бит #10 – Событие превышения верхнего порога ADC#3 или событие по высокому уровню на входе DIN#3

Бит #11 – Событие превышения верхнего порога ADC#4 или событие по высокому уровню на входе DIN#4

Бит #12 - Событие перехода нижнего порога вниз ADC#1 или событие по низкому уровню на входе DIN#1

Бит #13 - Событие перехода нижнего порога вниз ADC#2 или событие по низкому уровню на входе DIN#2

Бит #14 - Событие перехода нижнего порога вниз ADC#3 или событие по низкому уровню на входе DIN#3

Бит #15 - Событие перехода нижнего порога вниз ADC#4 или событие по низкому уровню на входе DIN#4

Данные о состоянии источника питания дополнительно передаются в служебных сообщениях протокола  
Следующие байты по порядку  
Младший байт идет первым (с меньшим номером)

Если Бит#0 Байт#1 = 1, то передается структура 7 байт – текущее время датчика.

Uint8 - Год 0-99

Uint8 - Месяц 1-12

Uint8 - День 1-31

Uint8 - Часы 0-23

Uint8 - Минуты 0-59

Uint8 – Секунды 0-59

Uint8 – День недели 1-7, 1 – понедельник

Если Бит#1 Байт#1 = 1, то передается температуры датчика DS18B20

Int16 с фиксированной точкой 0,1 град от -550 до +1250 Если датчик неисправен, то значение -32767.

Если Бит#2 Байт#1 = 1, то передается температуры датчика SI7021 (SHT21)

Int16 с фиксированной точкой 0,1 град от -550 до +1250 Если датчик неисправен, то значение -32767

Если Бит#3 Байт#1 = 1, то передается значение влажности датчика SI7021 (SHT21)

Int16 с фиксированной точкой 0,1 град от -550 до + от 0 до 999, что равно 0.0% - 99.9% Если датчик неисправен, то значение -32767.

Если Бит#4 Байт#1 = 1, то передается температуры датчика AM2302/DHT22

Int16 с фиксированной точкой 0,1 град от -550 до +1250 Если датчик неисправен, то значение -32767

Если Бит#5 Байт#1 = 1, то передается значение влажности датчика AM2302/DHT22

Int16 с фиксированной точкой 0,1 град от -550 до + от 0 до 999, что равно 0.0% - 99.9% Если датчик неисправен, то значение -32767.

Если Бит#6 Байт#1 = 1, то передается температуры датчика PT100

Int16 с фиксированной точкой 0,1 град от -2500 до +9000, что равно -250,0 + 9000,0 град. Если датчик неисправен, то значение -32767.

Если Бит#7 Байт#1 = 1, то передается значения канала ADC#1 в формате Int16. 0-1023

Если Бит#0 Байт#2 = 1, то передается значения канала ADC#2 в формате Int16. 0-1023

Если Бит#1 Байт#2 = 1, то передается значения канала ADC#3 в формате Int16. 0-1023  
 Если Бит#2 Байт#2 = 1, то передается значения канала ADC#4 в формате Int16. 0-1023  
 Если Бит#3 Байт#2 = 1, то передается состояние цифровых входов и реле. Битовый регистр Int16

Также передается, если включена передача любого счетчика:

Если Бит#4 Байт#2 = 1, Если Бит#5 Байт#2 = 1, Если Бит#6 Байт#2 = 1, Если Бит#7 Байт#2 = 1  
 - информация об ошибке NAMUR, если включен. Биты активны для соответствующих настроек порта

Бит 0 – состояние реле #1; 1- включено

Бит 1 – состояние реле #2

Бит 2 – состояние реле #3

Бит 3 – состояние реле #4

Бит 4 – Ошибка NAMUR #1

Бит 5 – Ошибка NAMUR #2

Бит 6 – Ошибка NAMUR #3

Бит 7 – Ошибка NAMUR #4

Бит 8 – состояние DIN#1;

Бит 9 – состояние DIN #2

Бит 10 – состояние DIN #3

Бит 11 – состояние DIN #4

Бит 12 – Состояние входа протечки #1; 1 – есть протечка

Бит 13 – Состояние входа протечки #2; 1 – есть протечка

Бит 14 – n/a

Бит 15 – n/a

Если Бит#4 Байт#2 = 1, то включена передача значения канала Счетчика #1 в формате uint32

Если Бит#5 Байт#2 = 1, то включена передача значения канала Счетчика #2 в формате uint32

Если Бит#6 Байт#2 = 1, то включена передача значения канала Счетчика #3 в формате uint32

Если Бит#7 Байт#2 = 1, то включена передача значения канала Счетчика #4 в формате uint32

### Download Command - Нисходящий канал

Длина посылки - переменной длины. Минимум 2 байта

Первый байт - функция - битовая маска, до 8 функций.

Второй байт - данные / команда для первой включённой функции, первой считается начиная с младшего бита.

Третий байт - данные / команда для первой включённой функции, и так далее.

1-Функция	данные / команда	Действие
Бит 0	0x00 -0x0F	Битовая маска – сброс показаний счетчиков
Бит 1	0x00 -0xFF	Битовая маска – управление выходы реле, по два бита на каждое реле 00- выкл; 01 – вкл; 10- изменить на противоположное; 11 - пульс

## Приложение 1

Парсер восходящих данных.

C#

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows.Forms;

namespace Data_Parser
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }

        private void button1_Click(object sender, EventArgs e)
        {
            try
            {
                textBox2.Text = "";
                string s = textBox1.Text;
                byte[] bytes = new byte[s.Length / 2];
                for (int i = 0; i < s.Length; i += 2)
                { bytes[i / 2] = Convert.ToByte(s.Substring(i, 2), 16); }
                // получили массив байт FE03000000C010A01021501EE00
                // Шапка
                DateTime dateTime = DateTime.Now;
                SprintLF( "Отчет от " + dateTime.ToString("HH:mm:ss") + " " + dateTime.ToString("dd-MM-yy") + " ");
                //Байт #0 - //0 означает внешнее питание, 1 - низкий уровень, 254 - высокий уровень и
                //255 означает, что конечное устройство не смогло измерить уровень заряда батареи.
                if (bytes[0] == 0) { SprintLF("Подключенно внешнее питание"); }
                else if (bytes[0] == 255) { SprintLF("Питание не контролируется"); }
                else
                {
                    int pow = bytes[0]; pow = pow * 100 / 254;
                    SprintLF ("Батарея = " + Convert.ToString(pow) + "%");
                }
                // получим битмассивы какие данные и какие события
                int Index = 5;

                short Status = 0;
                short Event = 0;
                short Value16 = 0;
                int Value = 0;
                int Dcel = 0;
                int Drob = 0;

                string str1 = "";
                string str2 = "";
                byte[] db = { 0, 0, 0, 0 };

                byte[] da = { bytes[1], bytes[2] }; Status = BitConverter.ToInt16(da, 0);
                da[0] = bytes[3]; da[1] = bytes[4]; Event = BitConverter.ToInt16(da, 0);
                //*****
                /// Парсер
                //*****
                if ((Status & 0x0001) > 0)
                {
                    str1 = Convert.ToString(bytes[Index]); Index++;
                    if (str1.Length < 2) { str1 = "0" + str1; }
                    str2 = str1 + ":";
                    str1 = Convert.ToString(bytes[Index]); Index++;
                    if (str1.Length < 2) { str1 = "0" + str1; }
                    str2 = str2 + str1 + ":";
                    str1 = Convert.ToString(bytes[Index]); Index++;
                    if (str1.Length < 2) { str1 = "0" + str1; }
                    str2 = str2 + str1 + " ";
                    str1 = Convert.ToString(bytes[Index]); Index++;
                    if (str1.Length < 2) { str1 = "0" + str1; }
                    str2 = str2 + str1 + "-";
                    str1 = Convert.ToString(bytes[Index]); Index++;
                    if (str1.Length < 2) { str1 = "0" + str1; }
                    str2 = str2 + str1 + "-20";
                    str1 = Convert.ToString(bytes[Index]); Index++;
                    if (str1.Length < 2) { str1 = "0" + str1; }
                    str2 = str2 + str1 + "#";
                    str1 = Convert.ToString(bytes[Index]); Index++;
                    str2 = str2 + str1;

                    SprintLF("Время контроллера : " + str2);
                }
            }
            //*****
        }
    }
}
```

```

if ((Status & 0x0002) > 0)
{
    da[0] = bytes[Index]; Index++; da[1] = bytes[Index]; Index++;
    Value16 = BitConverter.ToInt16(da, 0); Dcel = Value16 / 10; Drob = Value16 % 10;
    SprintLF("Датчик температуры 18B20; T = : " + Dcel.ToString() + "." + Drob.ToString() + " Град.С");
}
//*****
if ((Status & 0x0004) > 0)
{
    da[0] = bytes[Index]; Index++; da[1] = bytes[Index]; Index++;
    Value16 = BitConverter.ToInt16(da, 0); Dcel = Value16 / 10; Drob = Value16 % 10;
    SprintLF("Датчик температуры SI7021(SHT21) T = : " + Dcel.ToString() + "." + Drob.ToString() + " Град.С");
}
//*****
if ((Status & 0x0008) > 0)
{
    da[0] = bytes[Index]; Index++; da[1] = bytes[Index]; Index++;
    Value16 = BitConverter.ToInt16(da, 0); Dcel = Value16 / 10; Drob = Value16 % 10;
    SprintLF("Датчик влажности SI7021(SHT21) H = : " + Dcel.ToString() + "." + Drob.ToString() + " %");
}
//*****
if ((Status & 0x0010) > 0)
{
    da[0] = bytes[Index]; Index++; da[1] = bytes[Index]; Index++;
    Value16 = BitConverter.ToInt16(da, 0); Dcel = Value16 / 10; Drob = Value16 % 10;
    SprintLF("Датчик температуры AM2302/DHT22 T = : " + Dcel.ToString() + "." + Drob.ToString() + " Град.С");
}
//*****
if ((Status & 0x0020) > 0)
{
    da[0] = bytes[Index]; Index++; da[1] = bytes[Index]; Index++;
    Value16 = BitConverter.ToInt16(da, 0); Dcel = Value16 / 10; Drob = Value16 % 10;
    SprintLF("Датчик влажности AM2302/DHT22 H = : " + Dcel.ToString() + "." + Drob.ToString() + " %");
}
//*****
if ((Status & 0x0040) > 0)
{
    da[0] = bytes[Index]; Index++; da[1] = bytes[Index]; Index++;
    Value16 = BitConverter.ToInt16(da, 0); Dcel = Value16 / 10; Drob = Value16 % 10;
    SprintLF("Датчик температуры PT100 T = : " + Dcel.ToString() + "." + Drob.ToString() + " Град.С");
}
//*****
if ((Status & 0x0080) > 0)
{
    da[0] = bytes[Index]; Index++; da[1] = bytes[Index]; Index++;
    Value16 = BitConverter.ToInt16(da, 0);
    SprintLF("Значение АЦП#1 = : " + Value16.ToString() + " ед.");
}
//*****
if ((Status & 0x0100) > 0)
{
    da[0] = bytes[Index]; Index++; da[1] = bytes[Index]; Index++;
    Value16 = BitConverter.ToInt16(da, 0);
    SprintLF("Значение АЦП#2 = : " + Value16.ToString() + " ед.");
}
//*****
if ((Status & 0x0200) > 0)
{
    da[0] = bytes[Index]; Index++; da[1] = bytes[Index]; Index++;
    Value16 = BitConverter.ToInt16(da, 0);
    SprintLF("Значение АЦП#3 = : " + Value16.ToString() + " ед.");
}
//*****
if ((Status & 0x0400) > 0)
{
    da[0] = bytes[Index]; Index++; da[1] = bytes[Index]; Index++;
    Value16 = BitConverter.ToInt16(da, 0);
    SprintLF("Значение АЦП#4 = : " + Value16.ToString() + " ед.");
}
if ((Status & 0x0800) > 0)
{
    da[0] = bytes[Index]; Index++; da[1] = bytes[Index]; Index++;
    Value16 = BitConverter.ToInt16(da, 0);
    // битовый регистр
    if ((Value16 & 0x0001) > 0) {SprintLF("Реле #1 Включено ");}
    else { SprintLF("Реле #1 Выключено "); }

    if ((Value16 & 0x0002) > 0) { SprintLF("Реле #2 Включено ");}
    else { SprintLF("Реле #2 Выключено "); }

    if ((Value16 & 0x0004) > 0) { SprintLF("Реле #3 Включено ");}
    else { SprintLF("Реле #3 Выключено "); }

    if ((Value16 & 0x0008) > 0) { SprintLF("Реле #4 Включено ");}
    else { SprintLF("Реле #4 Выключено "); }

    if ((Value16 & 0x0010) > 0) { SprintLF("Ошибка NAMUR #1 "); }
    if ((Value16 & 0x0020) > 0) { SprintLF("Ошибка NAMUR #2 "); }
    if ((Value16 & 0x0040) > 0) { SprintLF("Ошибка NAMUR #3 "); }
    if ((Value16 & 0x0080) > 0) { SprintLF("Ошибка NAMUR #4 "); }

    if ((Value16 & 0x0100) > 0) { SprintLF("DIN #1 = Hi"); }
    else { SprintLF("DIN #1 = Lo"); }
    if ((Value16 & 0x0200) > 0) { SprintLF("DIN #2 = Hi"); }
    else { SprintLF("DIN #2 = Lo"); }
}

```

```

        if ((Value16 & 0x0400) > 0) { SprintLF("DIN #3 = Hi"); }
        else { SprintLF("DIN #3 = Lo"); }
        if ((Value16 & 0x0800) > 0) { SprintLF("DIN #4 = Hi"); }
        else { SprintLF("DIN #4 = Lo"); }

        if ((Value16 & 0x1000) > 0) { SprintLF("Протечка на датчике #1 "); }
        else { SprintLF("Нет протечки, датчик #1"); }
        if ((Value16 & 0x2000) > 0) { SprintLF("Протечка на датчике #2 "); }
        else { SprintLF("Нет протечки, датчик #2"); }

    }

    if ((Status & 0x1000) > 0)
    {
        // счетчик1
        db[0] = bytes[Index]; Index++; db[1] = bytes[Index]; Index++;
        db[2] = bytes[Index]; Index++; db[3] = bytes[Index]; Index++;
        Value = BitConverter.ToInt32(db, 0);
        SprintLF("Значение Счетчика №1 = : " + Value.ToString());
    }
    if ((Status & 0x2000) > 0)
    {
        // счетчик2
        db[0] = bytes[Index]; Index++; db[1] = bytes[Index]; Index++;
        db[2] = bytes[Index]; Index++; db[3] = bytes[Index]; Index++;
        Value = BitConverter.ToInt32(db, 0);
        SprintLF("Значение Счетчика №2 = : " + Value.ToString());
    }
    if ((Status & 0x4000) > 0)
    {
        // счетчик2
        db[0] = bytes[Index]; Index++; db[1] = bytes[Index]; Index++;
        db[2] = bytes[Index]; Index++; db[3] = bytes[Index]; Index++;
        Value = BitConverter.ToInt32(db, 0);
        SprintLF("Значение Счетчика №3 = : " + Value.ToString());
    }
    if ((Status & 0x8000) > 0)
    {
        // счетчик2
        db[0] = bytes[Index]; Index++; db[1] = bytes[Index]; Index++;
        db[2] = bytes[Index]; Index++; db[3] = bytes[Index]; Index++;
        Value = BitConverter.ToInt32(db, 0);
        SprintLF("Значение Счетчика №4 = : " + Value.ToString());
    }
    //*****
    // события
    //*****
    if ((Event & 0x0001) > 0) { SprintLF("Событие превышения порога по температуре"); }
    if ((Event & 0x0002) > 0) { SprintLF("Событие перехода нижнего порога по температуре"); }
    if ((Event & 0x0004) > 0) { SprintLF("Событие превышения порога по влажности"); }
    if ((Event & 0x0008) > 0) { SprintLF("Событие перехода нижнего порога по влажности"); }

    if ((Event & 0x0010) > 0) { SprintLF("Событие наличия воды датчика протечки #1 "); }
    if ((Event & 0x0020) > 0) { SprintLF("Событие наличия воды датчика протечки #2 "); }
    if ((Event & 0x0040) > 0) { SprintLF("Событие отсутствия воды датчика протечки #1 "); }
    if ((Event & 0x0080) > 0) { SprintLF("Событие отсутствия воды датчика протечки #2 "); }

    if ((Event & 0x0100) > 0) { SprintLF("Событие превышения верхнего порога ADC#1 или событие по высокому уровню на входе
DIN#1"); }
    if ((Event & 0x0200) > 0) { SprintLF("Событие превышения верхнего порога ADC#2 или событие по высокому уровню на входе
DIN#2"); }
    if ((Event & 0x0400) > 0) { SprintLF("Событие превышения верхнего порога ADC#3 или событие по высокому уровню на входе
DIN#3"); }
    if ((Event & 0x0800) > 0) { SprintLF("Событие превышения верхнего порога ADC#4 или событие по высокому уровню на входе
DIN#4"); }

    if ((Event & 0x1000) > 0) { SprintLF("Событие перехода нижнего порога вниз ADC#1 или событие по низкому уровню на входе
DIN#1"); }
    if ((Event & 0x2000) > 0) { SprintLF("Событие перехода нижнего порога вниз ADC#2 или событие по низкому уровню на входе
DIN#2"); }
    if ((Event & 0x4000) > 0) { SprintLF("Событие перехода нижнего порога вниз ADC#3 или событие по низкому уровню на входе
DIN#3"); }
    if ((Event & 0x8000) > 0) { SprintLF("Событие перехода нижнего порога вниз ADC#4 или событие по низкому уровню на входе
DIN#4"); }

    }
    catch
    {
        SprintLF("\r\n Ошибка входных данных\r\n");
    }
}
private void Sprint(string ss)
{
    textBox2.Text = textBox2.Text + ss;
}
private void SprintLF(string ss)
{
    textBox2.Text = textBox2.Text + ss + "\r\n";
}
}
}

```