

Функция универсальной интеграции датчиков BLE

Поддерживаемые устройства, руководство по настройке, описание формата

Введение

В данной инструкции мы хотим ознакомить пользователей с возможностью удобной интеграции любых BLE-датчиков с нашими приборами.

Для этого достаточно ознакомиться с протоколом производителя датчика, в котором рассылаются BLE-адверты.

Универсальная интеграция доступна на следующих устройствах:

Модель трекера	L3 L3R	A4	A4WIFI	A4 MAX	B2	C4
Версия ПО (не ниже)	[0.01]	[0.11]	[0.01]	[0.11]	[0.02]	[0.01]

Логика универсальной интеграции

Согласно установленным стандартам BLE, датчики транслируют в радио эфир открытые сообщения (адверты), представляющие из себя упорядоченный набор полей.

Структура полей состоит из трёх классов данных (LTV): длина, тип и данные поля.

Наиболее часто встречающиеся типы данных в адвертах, это:

Код	Значение
0x01	Flags (Флаги)
0xFF	Manufacturer Specific Data (Данные производителя)
0x09	Complete Local Name (Имя устройства)
0x06	Incomplete List of 128-bit Service Class UUIDs
0x07	Complete List of 128-bit Service Class UUIDs

Практика показывает, что полезную информацию производители датчиков размещают в поле "0xFF" – "Manufacturer Specific Data", после которого мы начинаем отсчёт байтов B1, B2 и т.д.

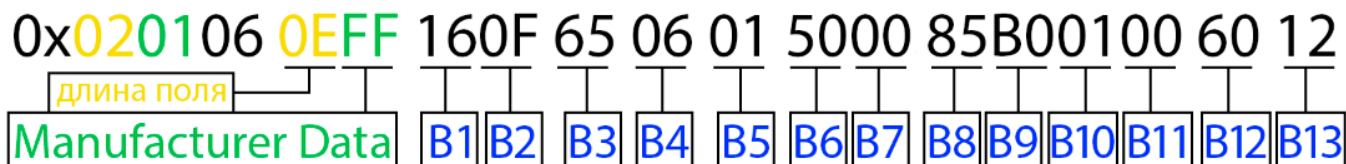


рис.1 - пример адверта

Примеры записи:

FF#V200 : [B1] – передача однобайтового параметра в виртуальном датчике

FF#V200 : [B1 , B2] – передача двухбайтового параметра в виртуальном датчике, способ big-endian

FF#V200 : [B2 , B1] – передача двухбайтового параметра в виртуальном датчике, способ little-endian

FF#V200:[B1, B2];FF#V201:[B3, B4] – добавлена передача второго двухбайтового параметра в виртуальном датчике

FF#V200:[B1, B2];FF#V201:[B3, B4];FF#T220:[B5, B6, B7, B8] – добавлена передача четырёхбайтового параметра в тэге



FF#V200:[B1, B2];FF#V201:[B3, B4];FF#T220:[B5, B6, B7, B8];V202:[R] – добавлена передача уровня сигнала в виртуальном датчике

Изменения! Добавлена настройка формирования внеочередных пакетов при изменении значения параметра на указанное число



Пример: FF#V200:[B1, B2];FF#V201:[B3, B4];FF#T220[B5, B6, B7, B8]!100;V202:[R] - формировать события при изменении значения параметра T220 каждые 100 единиц.

Примеры формирования строки

Часть записи	форма	Описание
Тип поля	FF	Код типа поля (в большинстве случаев это FF)
	#	Разделитель типа поля
Параметр для сервера	Vxxx	Номер виртуального датчика , в котором будет передан результат работы по формуле (до 3-х байт). Допустимые номера xxx = [200 .. 239] включительно, согласно нашему протоколу ☞ . В системе мониторинга соответствуют параметрам vs_200 .. vs_239
	Txxx	Номер тэга , в котором будет передан результат работы по формуле (до 4-х байт). Допустимые номера ууу = [220 .. 229] включительно, согласно нашему протоколу ☞ . В системе мониторинга соответствуют параметрам val_10 .. val_19  Важно! Назначаемые тэги не должны быть заняты другими датчиками или устройствами, например CAN-считывателем.
	:	Разделитель параметра сервера и набора данных адверта
Данные для передачи	[B1,B2,B3]	Набор данных адверта для записи в параметр для сервера - через запятую указываются номера байт данных адверта (см. рис.1 и примеры записи) и их порядок упаковки в виртуальный датчик или тэг.  Внимание! Производители датчиков используют разные способы хранения данных в памяти: Big-endian хранит старший байт числа по меньшему адресу памяти, что интуитивно понятно при чтении слева направо. Например, число 0x12345678 будет храниться, как 12 34 56

		<p>“ 78</p> <p>Little-endian делает обратное, помещая младший байт по меньшему адресу, так что то же число будет храниться, как 78 56 34 12</p> <p>В строке описания логики универсальной интеграции данные требуется записывать в порядке Little-endian</p>
	[R]	Уровень сигнала датчика для записи в параметр для сервера. Для данной записи не указывается тип поля.
Триггер по изменению значения	!xxx	<p>Символ " !" после квадратных скобок применяется для формирования внеочередного пакета при изменении значения параметра внутри этих скобок.</p> <p>Число после " !" указывает, на сколько должно измениться значение (дельта) для формирования внеочередного пакета, например: " !50 " - изменение более, чем на 50 единиц включительно.</p> <p>“ Данная форма не является обязательной.</p>
Разделитель между записями	;	Разделитель между формулами расчётов. В одном описании может быть до 5-ти формул.

“ **Внимание!** Максимальная длина строки для поля ввода описания 128 символов. Если необходимо внести более 5-ти формул или превышен лимит по символам в поле "**Универсальная интеграция датчиков**", то можно продолжить описание в тех же полях следующего датчика, оставив MAC-адрес прежним.

Пример 1

Рассмотрим беспроводной датчик **BASv6**, работающем в протоколе **BAS mini**

Пример пакета в шестнадцатеричном виде	Порядок байт	Обозначение	Диапазон данных	Описание
0EFF160F650601500085B001006012		0xFF = «Manufacturer data», длина = 0x0E	0x0EFF	
	[1,2]	Префикс производителя	0x160F	константа
	[3]	Тип датчика - датчик угла	0x65	константа
	[4]	Режим работы 0x06	0,4,5,6,9,10, 32	0 - склад 4 - перпендикулярное вращение 5 - параллельное вращение 6 - угол 9 - ковш 10 - отвал 32 - вибрация
	[5]	Событие сработки 0x01	0..2	В режимах вращения (параллельно и перпендикулярно): 0 - нет вращения 1 - вращение по часовой стрелке 2 - вращение против часовой стрелки В режиме вибрация: 0 - нет движения 1 - движение 2 - работа двигателя В остальных режимах: 0 - нет сработки 1 - есть сработка
	[6,7]	Количество полных оборотов со знаком Обороты в минуту двигателя Угол в градусах 0x0050 = 80°	-360...360	В режимах контроля вращения: полные обороты В режиме вибрация: текущие обороты двигателя в минуту В остальных режимах: угол в градусах
	[8,9,10, 11]	Моточасы 0x0001B085 или 110725 сек	0x00...0xFFFFFFFF	Время в эксплуатации (моточасы) в секундах для всех режимов
	[12]	Напряжение батареи в % 0x60 = 96%	0...100	
	[13]	Температура в градусах Цельсия 0x12 = 18° C	-128...127	

рис. 2 - описание Manufacturer Specific Data протокола BAS mini

Разберём на примере адверта этого датчика:

0x0201060EFF160F650601500085B001006012

Поля	Данные адверта	Значения для записи	Значение	
Служебные флаги	0x02	-	2 байта	Длина

	0x01	01#	Служебные флаги	Тип данных поля адверта
	0x06	[B1]	-	Значение
Данные производителя	0x0E	-	14 байт	Длина
	0xFF	FF#	Данные производителя	Тип поля адверта
	0x160F	[B1, B2]	3862	Код производителя
	0x65	[B3]	101 - BAS	Тип датчика
	0x06	[B4]	6 - Угол	Режим работы
	0x01	[B5]	1 - Работает	Событие сработки
	0x5000	[B6, B7]	80 °	Значение основного параметра в зависимости от режима работы
	0x85B00100	[B8, B9, B10, B11]	110725 сек	Моточасы
	0x60	[B12]	96 %	Напряжение батареи в %
	0x12	[B13]	18 °C	Температура в °C

Зная, где содержится нужная информация в адверте, с помощью **логики универсальной интеграции** мы можем передавать её на телематический сервер. Например, нам нужны только Моточасы, Температура и Угол.

Запись	Поле адверта		Параметр для записи		Значение из адверта	
Запись 1	FF#	Выбираем данные производителя	T221 :	Записываем в параметр va1_11	[B8, B9, B10, B11]	Значение моточасов
Запись 2	FF#		V201 :	Записываем в параметр vs_201	[B13]	Значение температуры
Запись 3	FF#		V202 :	Записываем в параметр vs_202	[B6, B7]	Значение угла

Значит, строка для записи в трекер будет иметь следующий вид:

FF#T221 : [B8, B9, B10, B11] ; FF#V201 : [B13] ; FF#V202 : [B6, B7]

Также необходимо указать MAC-адрес интегрируемого датчика

MAC адрес 1

5C5310D309AF ✓

[6 байт (A-F, 0-9)]

Универсальная интеграция датчиков

FF#T221:[B8,B9,B10,B11];FF#V201:[B13];FF#V202:[B6,B7] ✓

[128 байт (A-F, 0-9, #, :, [], :, {})]

рис. 3 - пример настройки устройства

В системе мониторинга мы видим переданные параметры:

vs_201: 18	vs_202: 80
val_11: 110725	

рис. 4 - переданные данные в систему мониторинга

Где:

val_11 : 110725 с – моточасы в секундах (1 день, 6 часов, 45 минут 25 секунд)

vs_201 : 18 °C – температура, °C

vs_202 : 80° – угол в градусах

Пример 2

Рассмотрим беспроводной датчик **TESLiOT One**, работающем в протоколе **TESLiOT protocol v.1.0**

Ниже представлен разобранный протокол, в котором работает данный датчик.

Data exchange is unidirectional and is performed via Advertisement broadcasts. Length of ADV-datagram is 28 bytes.

Field size, bytes	Field name	Type of content	Description	Example
7	Service data, do not analyze			
3	Type id	ASCII code	Device type ID	0x54 0x53 0x31 -> TS1
1	Version HW	Version, number	Hardware version	0x05 -> hw.5
1	Version SW	Version, number	Software version	0x0a -> sw.10
4	Unix-time	Seconds	Date, time, in POSIX format	0x5f 0x37 0xaa 0xdc -> 0x5f37 aadc -> sec=0 min=0 hour=12 day=15 month=8 year=20
1	Power	Volts, V	Power supply voltage multiplied by 10	0x21 -> 33 -> 3,3v
1	Collision Flags	STATUS, true/false	Event triggers: Door closed, magnetic sensor - 0x01 Door open, magnetic sensor - 0x02 Alert 1 - 0x04 Alert 2 - 0x08	0x0a -> 00001010 -> open door, alarm2
3	Accel	G-Sensor, g	Acceleration along axis X, Y, Z. Deduct 128 from this value, then divide by 100 to get actual data	0x8f 0x46 0x0a -> 143 70 10 -> X=0,15g Y=0,58g Z=-1,18g
1	Hall	Relative units	Magnetic field intensity	0x4a -> 74
2	Lux	Lux, lx	Light intensity	0x33 0xa4 -> 13220 -> 13220lx
2	Hum	Humidity, %	Humidity level. Divide this value by 100 to get actual data.	0x06 0x53 -> 16,19%
2	Temp	Temperature, °C	Temperature value, sign matters, divide by 100 to get actual data.	0xf9 0xca -> -15,90°C

рис. 5 - описание Manufacturer Specific Data протокола TESLiOT protocol v.1.0

Разберём на примере адверта этого датчика:

0x02010618FFFFFF5453310F0A6839BCDA1E0A808080000000A3F0988

Поля	Данные адверта	Значения для записи	Значение	
Служебные флаги	0x02	-	2 байта	Длина
	0x01	01#	Служебные флаги	Тип данных поля адверта
	0x06	[B1]	-	Значение

Данные производителя	0x18	-	24 байт	Длина
	0xFF	FF#	Данные производителя	Тип поля адверта
	0xFFFF	[B2, B1]	65535	Код производителя
	0x545331	[B5, B4, B3]	TS1	Тип датчика
	0x0F	[B6]	15	Версия аппаратного обеспечения
	0x0A	[B7]	10	Версия ПО
	0x6839BCDA	[B11, B10, B9, B8]	30.05.2025 14:12:42 (GMT+00)	Системное время датчика
	0x1E	[B12]	3 В	Напряжение батарейки
	0x0A	[B13]	дверь открыта	Статусы дверей
	0x80	[B14]	0 g	Ось X
	0x80	[B15]	0 g	Ось Y
	0x80	[B16]	0 g	Ось Z
	0x00	[B17]	0 А/м	Напряженность магнитного поля
	0x0000	[B19, B18]	0 lx	Освещённость
	0x0A3F	[B21, B20]	26,23 %	Влажность
	0x0988	[B23, B22]	24,4 °C	Температура

Допустим, мы хотим передавать на сервер мониторинга версию платы датчика, напряжение батарейки, температуру, влажность:

Запись	Поле адверта		Параметр для записи		Значение из адверта	
Запись 1	FF#	Выбираем данные производителя	V205:	Записываем в параметр vs_205	[B6]	Версия аппаратного обеспечения
Запись 2	FF#		V206:	Записываем в параметр vs_206	[B12]	Напряжение батарейки
Запись 3	FF#		V227:	Записываем в параметр	[B23, B22]	Температура

			vs_227		
Запись 4	FF#	V226 :	Записываем в параметр vs_226	[B21 , B20]	Влажность



Внимание! В данном случае требуется записывать байты в обратном порядке для корректного отображения на сервере мониторинга, так как производитель датчика использовал способ **big-endian**.

Строка для записи в трекер будет выглядеть так:

FF#V205:[B6];FF#V206:[B12];FF#V227:[B23,B22];FF#V226:[B21,B20]

Также необходимо указать MAC-адрес интегрируемого датчика

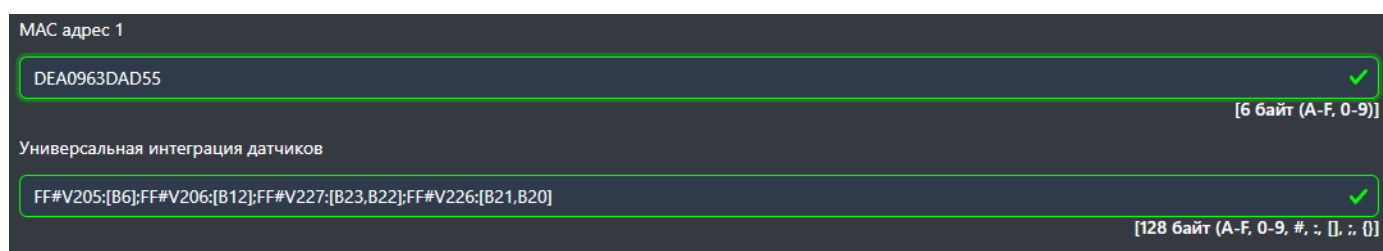


рис. 3 - пример настройки устройства

На сервере мониторинга Wialon мы видим пришедшие параметры:

vs_205: 15	vs_206: 30
vs_226: 2623	vs_227: 2440

рис. 4 - переданные данные в систему мониторинга

Где:

vs_205 : 15 – версия платы датчика

vs_206 : 30 = 3,0 В – напряжение батарейки (значение необходимо разделить на 10 согласно протоколу производителя)

vs_226 : 2623 = 26,23 % – влажность, % (значение необходимо разделить на 100 согласно протоколу производителя)

vs_227 : 2440 = 24,4 °С – температура, °С (значение необходимо разделить на 100 согласно протоколу производителя)