ViPen. Bluetooth протокол

v2.06 16.12.2022
AndreySchekalev@vibrocenter.ru

1. Данные в беконе Advertisement

```
typedef __packed struct stAdvertisingData
{
	uint8_t Len1;
	uint8_t Type1;
	uint8_t Flag1;

	uint8_t Len9;
	uint8_t Type9;
	char Name9[5];

	uint8_t LenFF;
	uint8_t TypeFF;
	uint16_t ManID; //  (Texas Instruments Inc. = 000D)
	uint8_t Addr;    // ==0
	uint16_t ID;      // const uint16_t ViPen_Bt_ID     = 0x4F5C; // Magic number
	uint32_t TimeStamp;     // Счётчик 1024 Гц для проверки, что появились новые данные
	int16_t Values[4];         // Velocity, Acceleration, Excess, Temperature
} TAdvertisingData;
#define szTAdvertisingData sizeof(TAdvertisingData)
static_assert(szTAdvertisingData==29, "");
```

```
#define GAP_ADTYPE_FLAGS                1

#define GAP_ADTYPE_LOCAL_NAME_COMPLETE     9

#define GAP_ADTYPE_MANUFACTURER_SPECIFIC    0xFF


#define GAP_ADTYPE_FLAGS_GENERAL    0x02

#define GAP_ADTYPE_FLAGS_BREDR_NOT_SUPPORTED    0x04


#define TI_COMPANY_ID (0x000D) // Company Identifier: Texas Instruments Inc. (13)

#define ViPen_Bt_ID   (0x4F5C) // Magic number


// бекон

static TAdvertisingData AdvertisingData =

{

        0x02, GAP_ADTYPE_FLAGS, GAP_ADTYPE_FLAGS_GENERAL |
GAP_ADTYPE_FLAGS_BREDR_NOT_SUPPORTED,

        0x06, GAP_ADTYPE_LOCAL_NAME_COMPLETE, 'V', 'i', 'P', 'e', 'n',

        0x12, GAP_ADTYPE_MANUFACTURER_SPECIFIC, TI_COMPANY_ID,

        0x00, ViPen_Bt_ID, 0x00000000, 0, 0, -200, 0 // User_Data (п.2)

};
```

2. Структура User_Data:

15 байт, как в данных бекона. Первый байт адреса == 0.
Порядок байт в int16_t – Low8, High8 (little-endian, интеловский)
Характеристика 3890BE9F3A5E459DB799102365770001

```
typedef __packed struct stUser_Data

{

        uint8_t Addr;    // ==0

        uint16_t ID;      // const uint16_t ViPen_Bt_ID     = 0x4F5C; // Magic number

        uint32_t Timestamp;     // Счётчик 1024 Гц для проверки, что появились новые данные

        int16_t Values[4];         // Velocity, Acceleration, Excess, Temperature

} TUser_Data;
```

Velocity = Виброскорость, СКЗ 10..1000Гц, мм/с, *100, (7,1 мм/с = 0x02C6)

Acceleration = Виброускорение, Пик, м/с², *100, (4,5 м/с² = 0x01C2)

Excess = Эксцесс ускорения, о.е., *100, (0,1 = 0x000A, -2,0 = 0xFF38) – для диагностики подшипников

Temperature = Температура, град С, *100, (28,3° = 0x0B0E, -10,0° = 0xFC18)

3. Данные в Private service

Сервис
(Private service UUID):

378B4074C2B845FF894C418739E60000
378B4074-C2B8-45FF-894C-418739E60000

Характеристика для приёма User_Data (п.2)
(Private characteristic UUID):

Read, Notify, 15 байт

3890BE9F3A5E459DB799102365770001
3890BE9F-3A5E-459D-B799-102365770001

Характеристика для управления прибором (п.4)
(Private characteristic UUID):

Write, Read, Notify, 2 байта

3890BE9F3A5E459DB799102365770002
3890BE9F-3A5E-459D-B799-102365770002

Характеристика для передачи запроса сигнала (п.5)
(Private characteristic UUID):

Write, 2 байта

3890BE9F3A5E459DB799102365770003
3890BE9F-3A5E-459D-B799-102365770003

Характеристика для приёма сигнала Waveform_Data (п.5)
(Private characteristic UUID):

Indicate, 150 байт

3890BE9F3A5E459DB799102365770004
3890BE9F-3A5E-459D-B799-102365770004

4. Управление прибором

Порядок байт в uint16_t – Low8, High8 (little-endian, интеловский)

Команда, Write в характеристику 3890BE9F3A5E459DB799102365770002:

const uint16_t ViPen_Command_Start = 0x0001; // Запустить измерение (зажать кнопку)

const uint16_t ViPen_Command_Stop = 0x0002; // Остановить измерение (отпустить кнопку)

const uint16_t ViPen_Command_Off = 0x0003; // Выключить прибор

const uint16_t ViPen_Command_Idle = 0x0004; // Чтобы прибор не выключался через 1 минуту, иногда будить его

Биты состояния, Read, Notify:

```
const uint16_t ViPen_State_Stoped      = (0<<0); // Прибор стоит
const uint16_t ViPen_State_Started     = (1<<0); // Прибор в режиме измерения (может быть с кнопки)
const uint16_t ViPen_State_NoData      = (0<<1); // Данных нет (после инициализации)
const uint16_t ViPen_State_Data        = (1<<1); // Есть данные
```

// Эти команды только для производителя прибора. Пользователю их не показывать

```
#define ViPen_Command_Calibration   0x0010 // Вход в режим тест/калибровка без переключателя
```

5. Передача сигнала

Команда, Write в характеристику 3890BE9F3A5E459DB799102365770003:

```
const uint16_t ViPen_Get_Data_Vel          = 0x0010; // запросить сигнал Канала Velocity
const uint16_t ViPen_Get_Data_Acc          = 0x0011; // запросить сигнал Канала Acceleration
```

6. Структура Waveform_Data:

150 байт
Порядок байт в int16_t – Low8, High8 (little-endian, интеловский)
Характеристика 3890BE9F3A5E459DB799102365770004

```
#define DATA_BLOCK_LEN (150) // длина блока данных
#define STAMPS_IN_BLOCK ((DATA_BLOCK_LEN-2)/2) // 74 отсчёта в блоке
```

Блок 0 – заголовок:

```
typedef __packed struct stWaveform_ Header
{
        uint8_t ViPen_Get_Data_Command;     // Команда из п.5
        uint8_t ViPen_Get_Data_Block;       // Номер блока == 0
        uint8_t ViPen_Get_Wave_ID;    // Счётчик, позволяет проверить, что качаем тот-же замер
                                      // Увеличивается на 1, при запросе заголовка
        uint8_t Reserv1;
        uint32_t Timestamp;     // Счётчик 1024 Гц, совпадает с User_Data. Timestamp
```

```
        float Coeff;       // Коэф перевода данных int16_t  в float, 4 байта
        unt16_t Reserv2[ (DATA_BLOCK_LEN-12)/2];
} TWaveform_ Header;
#define szTWaveform_Header sizeof(TWaveform_Header)
static_assert(szTWaveform_Header==DATA_BLOCK_LEN, "");
```

Блок N – данные:

```
typedef __packed struct stWaveform_Data =
{
        uint8_t ViPen_Get_Data_Block;          // Номер блока. Обычно идут по-порядку (1..22)
        uint8_t ViPen_Get_Wave_ID;    // Счётчик, позволяет проверить, что качаем тот-же замер
        int16_t Wave[STAMPS_IN_BLOCK];       // отсчёты = 2 байта знаковое * 74 отсчёта в блоке
} TWaveform_Data;
#define szTWaveform_Data sizeof(TWaveform_Data)
static_assert(szTWaveform_Data==DATA_BLOCK_LEN, "");
```

Всего в сигнале 1600 отсчётов по 2 байта = 22 блока + 1 заголовочный по 150 байт.
Лишние отсчёты (1600-ый и дальше) == 0 – их не учитывать.
Частота дискретизации = 4кГц = 0.25мс
Длительность сигнала = (1600-1)*0.25=399.75мс

7. Работа с прибором

Пока нажата кнопка, передаются данные о вибрации в беконе Advertisement (п.1) и обновляются эти-же данные в Характеристика для передачи User_Data (п.2).

Обновились ли данные, можно смотреть по полю Timestamp. Если Timestamp==0 – данных ещё нет.

Проверять поле ViPen_Bt_ID == 0x4F5C // Magic number

Читать состояние прибора и управлять Старт-Стоп – п.4.

Чтобы прибор не выключался через 1 минуту, раз в 30 секунд будить его командой ViPen_Command_Idle

Запрос сигнала (п.5). Write команду:

ViPen_Get_Data_xx – номер канала = Vel / Acc

Получение сигнала (п.6):

Прибор посылает 23 блока по 150 байт по Indicate.

Блок 0 – заголовок TWaveform_ Header

1..22 – данные TWaveform_Data

Смотреть по полю Hdr.ViPen_Get_Wave_ID == Block.ViPen_Get_Wave_ID, что сигнал не обновился во время передачи

Смотреть блоки по полю ViPen_Get_Data_Block, что пришли все блоки