

# Радиомодуль nanoNET с интерфейсом USB



## Инструкция пользователя

## Содержание

Содержание.....	2
Описание демонстрационного набора USB-nanoNET .....	3
Общие технические характеристики радиомодулей .....	3
Возможные сферы применения .....	3
Техническое описание радиомодуля USB-nanoNET .....	4
Общая принципиальная схема устройства .....	4
Блок-схема радиомодуля .....	4
Принцип работы радиомодуля .....	5
Программирование радиомодуля USB-nanoNET .....	6
Список компонентов.....	7
Расположение компонентов.....	8
Программное обеспечение .....	9
USB драйвер FTDI .....	9
Демонстрационное программное обеспечение .....	10
Общие вопросы компиляции проекта .....	10
Программа проверки качества связи.....	11
Выбор конфигурации проекта (MASTER SLAVE).....	12
Выбор конфигурации проекта (устройство вывода информации USB USART).....	12
Выбор режимов работы трансиверов.....	12
Настройка RS-232 .....	13

## Описание демонстрационного набора USB-nanoNET

Демонстрационный набор из двух радиомодулей USB-nanoNET может быть использован для организации полудуплексного беспроводного соединения со скоростью обмена данными около 700 Кбит/сек на расстояниях до 200 метров (до 900 метров при использовании направленных антенн).

Подключение модулей к компьютеру реализовано с помощью интерфейса USB. Встроенное программное обеспечение организует работу устройств в режиме Master-Slave. Демонстрационный набор предназначен для тестирования и оценки качества радиосвязи (в том числе в режиме sniffера). Может также использоваться разработчиками в качестве отладочного средства с возможностью смены прошивки микроконтроллера. Управление процессом обмена данными может выполняться с помощью любой терминальной программы на персональном компьютере (например, Hyper Terminal).

Дополнительные характеристики:

- ❖ антенный разъем SMA-female
- ❖ программирование управляющего микроконтроллера ATmega32L по интерфейсу SPI
- ❖ возможность подключения автономного электропитания
- ❖ использование внешней памяти типа FRAM (опционально)
- ❖ использование микросхемы со статическим MAC-адресом (опционально)

### **Общие технические характеристики радиомодулей**

- Используемые трансиверы – nanoPAN 5361 (производства Nanotron Technologies GmbH)
- Битовые скорости передачи 1 Мбит/сек и 2 Мбит/сек
- Уникальный метод частотного chirp-кодирования
- Малое энергопотребление, эффективный энерговклад
- Возможность функционирования с привязкой к универсальному времени UTC
- Совместимость со стандартом IEEE 802.15.4a
- Безлицензионный диапазон ISM 2.4ГГц

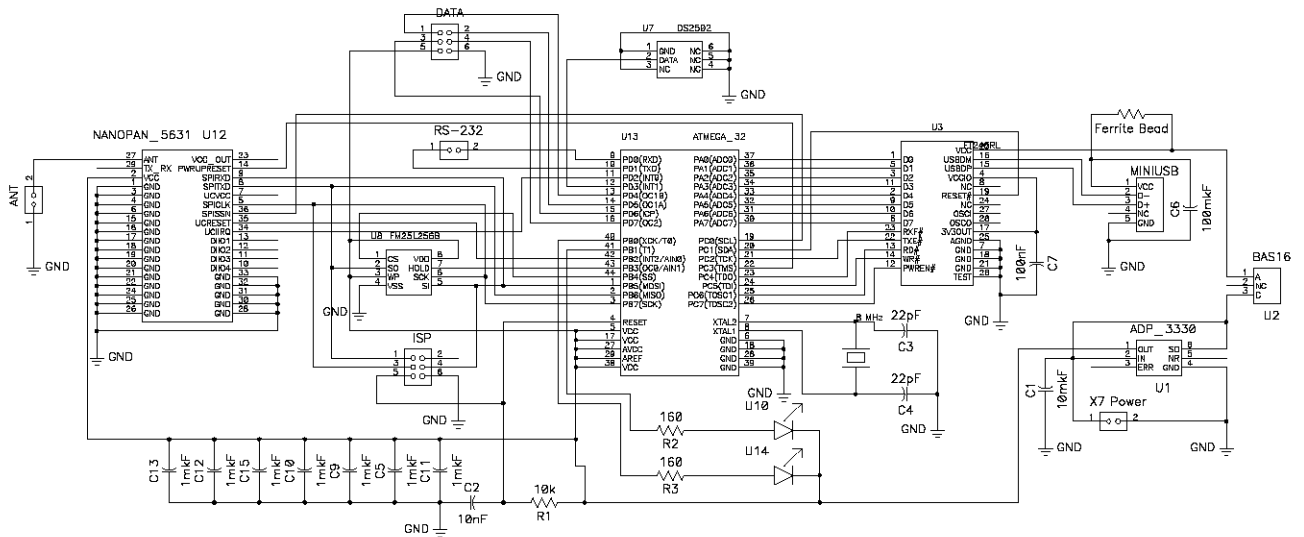
Использование модулей на территории Российской Федерации регламентировано в постановлениях ГКРЧ РФ (например, смотри [http://lab127.ru/rf\\_license.html](http://lab127.ru/rf_license.html)).

### **Возможные сферы применения**

- Беспроводные сети датчиков
- Технологический промышленный контроль и мониторинг
- Медицина и защита окружающей среды
- Радиочастотные метки RFID / мониторинг движения мобильных объектов
- Безопасность
- Сети сбора распределенной информации
- Сейсмоконтроль
- Технология "умный дом"
- Радиосвязь в критических по энергопотреблению приложениях

# Техническое описание радиомодуля USB-nanoNET

## Общая принципиальная схема устройства

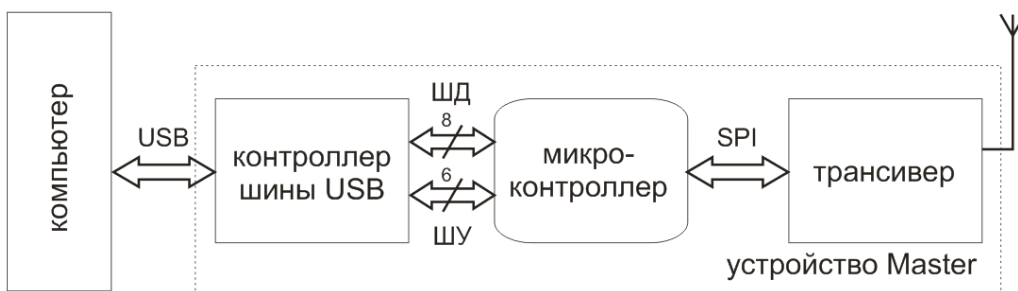


## Блок-схема радиомодуля

Устройство подключается к компьютеру по интерфейсу USB и состоит из следующих блоков:

- управляющий микроконтроллер (ATmega32L);
- трансивер стандарта nanoNET (nanoPAN 5361);
- контроллер шины USB (FT245RL).

На рисунке ниже приведена функциональная блок-схема устройства (расположение блоков слева направо обратно по отношению к порядку на принципиальной схеме).



## Принцип работы радиомодуля

Питание устройства осуществляется по шине USB (+5 В) стабилизатором ADP3330 (U1), формирующим шину с напряжением +3 В (для питания ATmega32L и nano PAN5361). Кроме этого возможно подключение внешнего питания (+3.5 .. +5-6 В) к разъему X7 (например в режиме программирования микроконтроллера ATmega32L).

**Не следует** подключать одновременно оба источника питания (через USB шину и через разъем X7), это может вызвать выход из строя устройства.

Конденсаторы C5, C9, C10, C11, C12, C13, C15 емкостью 0.1 .. 1 мкФ – блокировочные. Ферритовое кольцо (Ferrite\_Bead) и конденсатор C6 образуют LC цепочку и необходимы для уменьшения проникновения ВЧ наводок в компьютер по шине USB.

Управляющий микроконтроллер ATmega32L (U13) тактируется с помощью цепи из кварцевого резонатора на 8 МГц (8 MHz) и двух возбуждающих колебания конденсаторов C3 и C4 емкостью по 22 пФ. RC цепочка (R1 номиналом 10 кОм, C2 номиналом 10 нФ) формирует импульс сброса в момент включения питания для корректного запуска микроконтроллера.

Светодиоды U10, U14 (рядом с ограничивающими ток резисторами R2, R3) можно использовать, например, для индикации режима приема-передачи устройства или обозначения факта обмена данными по шине USB. (Внимание: светодиоды могут отсутствовать на вашей плате или быть подключены к общей шине, на принцип. схеме они подключены к шине питания).

Линии PC0, PC3, PB2, PB5, PB6, PB7, PD2 формируют шину управления/данных, связывающую микроконтроллер ATmega32L и трансивер nanoPAN 5361 (линии SPISSN, PWRUPRESET, UCRESET, SPIRXD, SPITXD, SPICLK, UCIRQ соответственно). В данной модификации программного обеспечения, поставляемого с устройством, линия PD2 (прерывание от трансивера) не используется; осуществлен программный опрос готовности приемо-передатчика.

Линии PB5, PB6, PB7, RESET, GND также выведены на разъем ISP для внешнего программирования ATmega32L по интерфейсу SPI (линия RESET в этом случае выступает в качестве SPISSN).

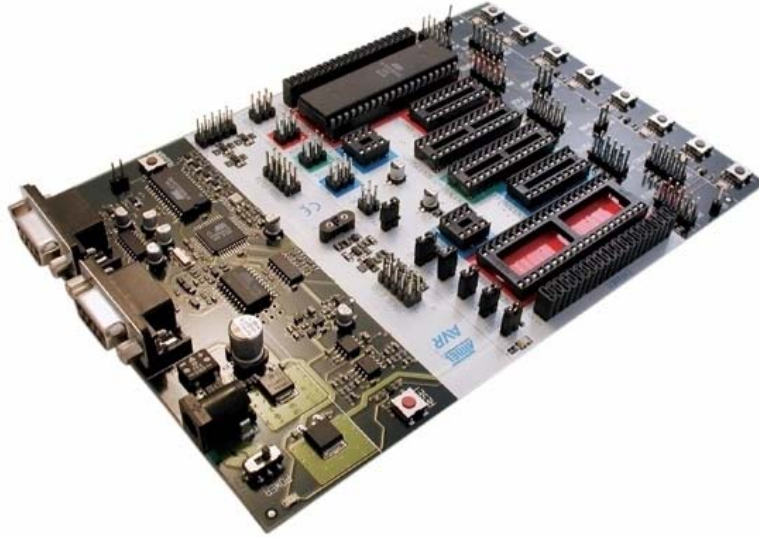
Интерфейс RS-232 (и соответствующий разъем) служит для отладочных целей. необходимо отметить, что для подключения к компьютеру необходимо использовать дополнительную микросхему формирователя уровней, например MAX232, или воспользоваться формирователем, встроенным в некоторые программаторы (например, STK-500).

8 цифровых линий порта А (шина данных ШД 8 на блок-схеме) и 6 линий порта С (все, кроме PC0 и PC3, ШУ 6 на блок-схеме) используются для обмена данными с микросхемой контроллера шины USB FT245RL, питающейся непосредственно от шины USB через упомянутую LC цепь.

FT245RL (производитель FTDI chip) представляет собой практически законченный модуль для сопряжения с микроконтроллером с преобразованием данных из формата USB в параллельный формат.

## Программирование радиомодуля USB-nanoNET

Программирование радиомодуля осуществляется с помощью внешнего программатора по интерфейсу SPI. Удобно использовать отладочный комплект STK-500 (см. рисунок), поскольку контакты разъема ISP6PIN (STK-500, с краю платы посередине длинной верхней стороны на рисунке) полностью совпадают по расположению с контактами разъема ISP на радиомодуле.



### Процедура программирования радиомодуля USB-nanoNET

1. Отладить и откомпилировать исходный код программы (в результате сгенерируется hex-файл).
2. Отключить питание от радиомодуля.
3. Подключить радиомодуль к программатору (например, шлейфом из 6 проводов).
4. Подключить внешнее питание через разъем X7 Power соблюдая полярность (питание от шины USB отключено). Несоблюдение полярности чаще всего приводит к выходу из строя стабилизатора ADP3330.
5. Опознать программным обеспечением микроконтроллер (ATmega32L или ATmega32) и убедиться, что логические переключки (fuses) выставлены **правильно** (запуск от внешнего кварцевого резонатора высокой частоты и отключение интерфейса JTAG, поскольку линии порта B используются для взаимодействия с чипом FT245RL).
6. Запрограммировать радиомодуль.
7. Отключить питание через разъем X7 Power, если модуль планируется подключать к шине USB.

Если ATmega32L программируется с ошибками, попробуйте уменьшить тактовую частоту ISP до 115 кГц и ниже (например, параметр ISP Freq во вкладке Board окна программирования в AVRStudio 4).

**Список компонентов**

Наим.	Номинал	Корпус	Описание	Обязательный или нет (Yes\No)
C1	10 mkF, 16V		Electrolytic capacitor	Y
C2	10 nF	1206	Capacitor	Y
C3	22 pF	1206	Capacitor	Y
C4	22 pF	1206	Capacitor	Y
C5	1 mkF	1206	Capacitor	Y
C6	100 mkF, 16V		Electrolytic capacitor	Y
C7	100 nF	1206	Capacitor	Y
C9	1 mkF	1206	Capacitor	Y
C10	1 mkF	1206	Capacitor	Y
C11	1 mkF	1206	Capacitor	Y
C12	1 mkF	1206	Capacitor	Y
C13	1 mkF	1206	Capacitor	Y
C15	1 mkF	1206	Capacitor	Y
R1	10 kΩ	1206	Resistor	Y
R2	160 Ω	1206	Resistor	N, for LED
R3	160 Ω	1206	Resistor	N, for LED
R10	Ferrite Bead MI0805K400R-00	0805	Ferrite Bead	Y
U1	ADP3330ART-3	SOT-23	Power conditioner (3V)	Y
U2	BAS16	SOT-23	Diode	Y
U3	FT245RL	SSOP-28	USB FIFO	Y
U4	8 MHz	<a href="http://crystekcrystals.com/crystal/spec-sheets/crystal/CYSDxx.pdf">http://crystekcrystals.com/crystal/spec-sheets/crystal/CYSDxx.pdf</a>	Quartz	Y
U5	2-pin connector		RS-232 connection	N
U6	6-pin connector		ISP* connector	Y
U7	DS2502	TSOC 6	External MAC address	N
U8	FM25L256B	SOIC 8	FRAM Serial	N, only in variant with data storage
U10	LED (any)	1206	Green LED	N
U11	3-pin connector	SMA female	Antenna connection	Y
U12	nanoPAN 5361			Y
U13	ATMega32L-8AU	TQFP 44	Microcontroller	Y
U14	LED (any)	1206	Red LED	N
U15	6-pin connector		Data connection	N
U16	7-pin connector (8969-B05C30DBA)	<a href="ftp://ftp.efo.ru/pub/cypress/usb/8969D02001-A3-B05CDB.pdf">ftp://ftp.efo.ru/pub/cypress/usb/8969D02001-A3-B05CDB.pdf</a>	miniUSB	Y
X7	2-pin connector		Power (4-12V DC)	Y, for SPI programming

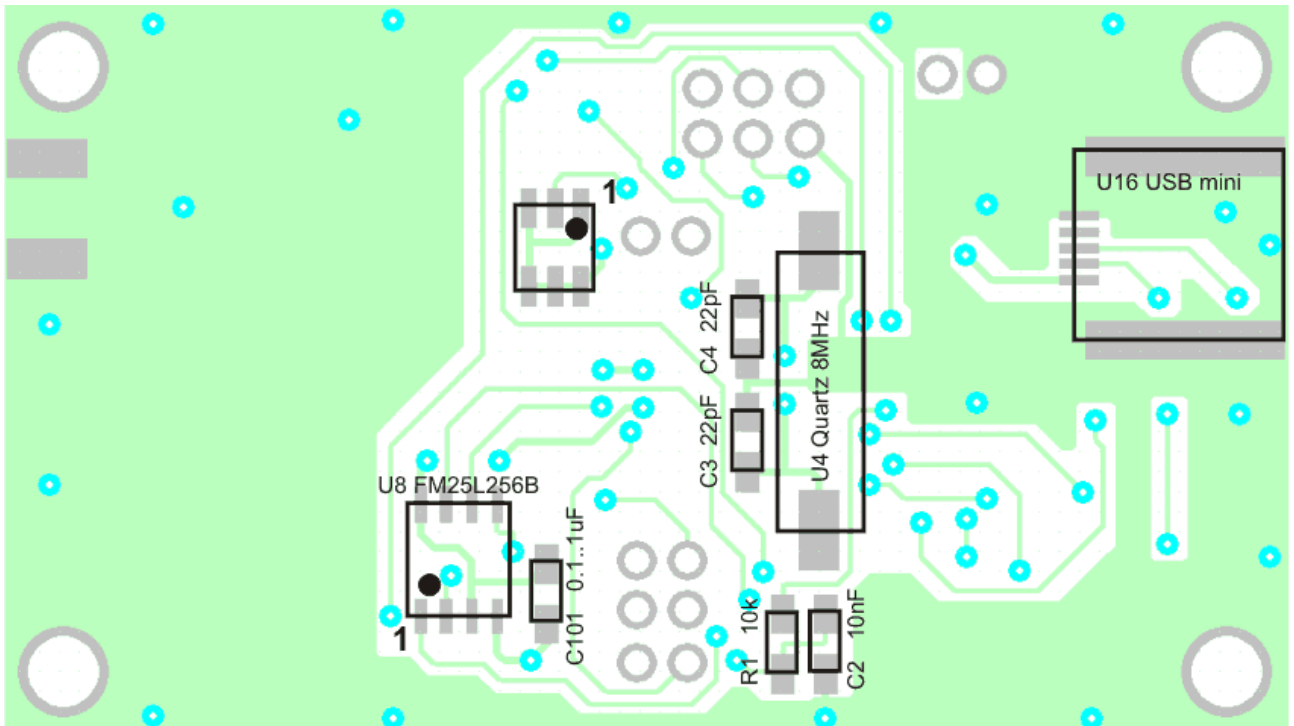
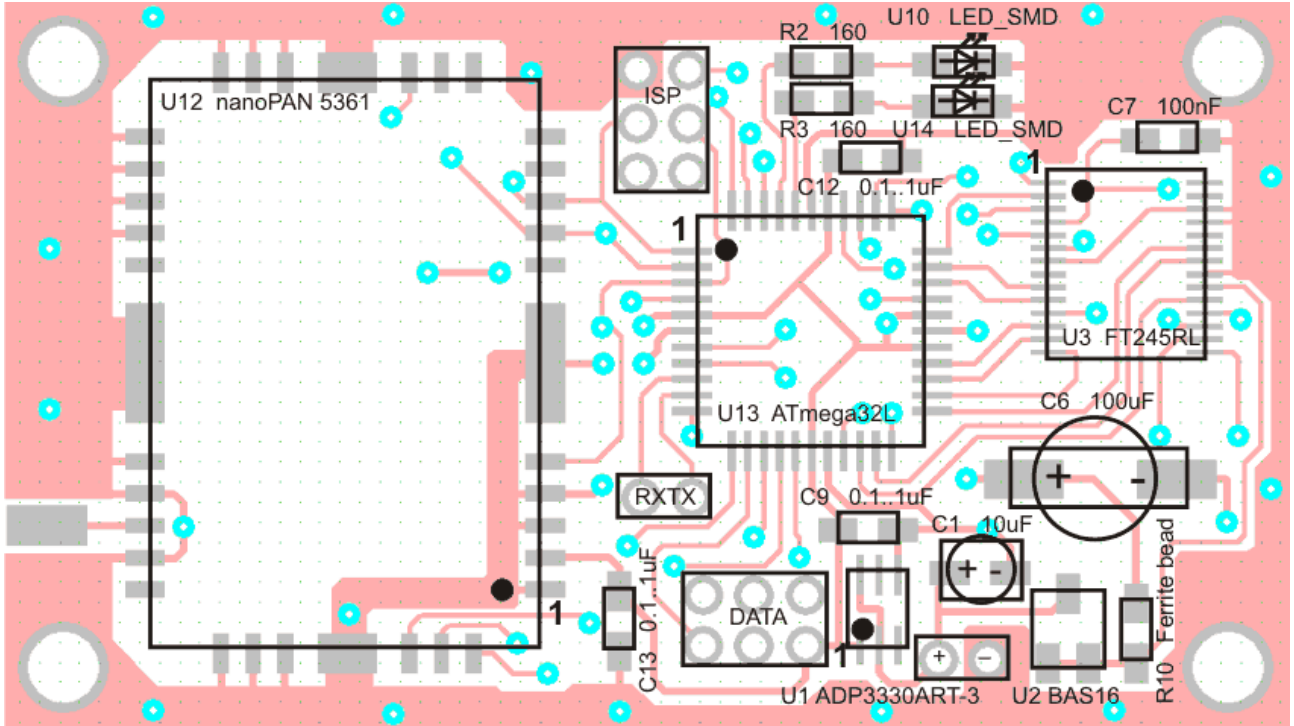
\*ISP – In System Programming

## Расположение компонентов

### Внимание!

Ваша плата и/или расположение элементов на ней может немного отличаться от представленного на рисунках ниже.

Например, не установлены светодиоды U10 и U14 и токоограничивающие резисторы R2 и R3, или светодиоды U10, U14 могут быть подключены не на общую шину (земля), а к шине питания (на рисунке они подключены к общей шине, правая часть светодиодов заземлена).



## Программное обеспечение

### **USB драйвер FTDI**

Для работы устройства требуется предварительно установить на компьютер драйвера, позволяющие взаимодействовать с чипом FT245RL. Их можно скачать с сайта производителя: <http://ftdichip.com/Drivers/VCP.htm>.

Для операционной системы Windows XP выпущен сертифицированный компанией Microsoft драйвер (<http://ftdichip.com/Drivers/CDM/CDM%202.04.06%20WHQL%20Certified.zip>).

Драйвер для операционных систем семейства Windows позволяет интерпретировать микросхему, подключенную к USB, либо как COM-порт, либо обращаться к ней с использованием вызовов драйвера.

После установки драйверов (!) при подключении радиомодуля с помощью USB кабеля (!!) в системе должно появиться новое устройство. Убедиться в этом можно, кликнув правой клавишей по ярлыку "Мой компьютер", далее выбрать "Свойства системы" / "Оборудование" / "Диспетчер устройств" / "Порты COM и LPT". Если радиомодуль не подключен к компьютеру, новый COM-порт в системе обнаружен не будет.

Поскольку драйвера FTDI позволяют работать с чипом FT245RL в режиме последовательного порта, для общения с платой можно использовать любую терминальную программу, например Гипертерминал из стандартной поставки Windows.

Причем в параметрах конфигурации необходимо лишь правильно выбрать номер COM-порта; остальные настройки (битовая скорость, формат данных, управление потоком) можно оставить по умолчанию. Обмен данными полностью контролируется драйвером FTDI, выбор конкретных настроек COM-порта не влияет на режим работы.

### **Удаление неиспользуемых устройств, занимающих COM-порты с низкими номерами**

В некоторых случаях в операционной системе Windows может понадобиться освободить низкие номера COM-портов, которые были зарегистрированы под ранее использовавшееся оборудование.

Для этого запустить апплет "Панель управления" / "Система" / "Дополнительно" / "Переменные среды", создать переменную DEVMGR\_SHOW\_NONPRESENT\_DEVICES и присвоить ей значение 1. После этого закрыть окно Система.

Эта переменная сделает активным пункт меню "Показать скрытые устройства" в диспетчере устройств. Теперь снова необходимо запустить "Панель управления" / "Система" / "Оборудование" / "Диспетчер устройств" и выставить галочку в меню "Вид" / "Показать скрытые устройства".

Ненужные из неактивных устройств в секции "Порты (COM и LPT)" можно удалить.

## Демонстрационное программное обеспечение

### Общие вопросы компиляции проекта

С платой может поставляться демонстрационное программное обеспечение для двух разных сред разработчика (и, соответственно, разных компиляторов языка C): CodeVisionAVR и WinAVR (надстройка над AVRStudio 4). Оба пакета поддерживают работу с программатором STK-500.

Коллектив разработчиков рекомендует пользоваться CodeVisionAVR из-за большего удобства в работе и более эффективной сборки исполняемого модуля благодаря встроенным по умолчанию возможностям использования флеш памяти микроконтроллера для хранения констант (программное обеспечение nanoNET содержит большое количество таблиц и строковых констант, их невыгодно располагать в оперативной памяти). Кроме этого, в CodeVisionAVR есть встроенный терминал, позволяющий визуализировать на экране выходной поток, формируемый, например, функцией printf(), и сохранять полученные данные в файл.

Далее следует описание исходного текста программ для компилятора avr-gcc, входящего в пакет WinAVR.

Программное обеспечение для радиомодулей построено на основе демонстрационного драйвера TRXdd2.01 для системы nanoNET.

В него входят следующие файлы.

appl_functions.c	Высокоуровневые функции
cchirpseq.c	Массив инициализации для формирования чирп-импульсов
dilcom.c	Описание структур
ft245_functions.c	Функции для работы с шиной USB
parallel_port_access.c	Низкоуровневые аппаратно-зависимые функции
nnreset.c	Сброс трансивера
nnspi.c	Функции транзакций по шине SPI
nn_usb_link_test.c	Файл, содержащий main()
ntrxinit.c	Инициализационные таблицы констант и функции
ntrxdil.c	Функции работы с трансивером
ntrxutil.c	Таблицы констант и утилиты для работы с трансивером

Кроме этого в подкаталоге /h содержатся заголовочные файлы.

Необходимо отметить, что перенаправление ввода-вывода в коде для avr-gcc и для CodeVisionAVR реализуется по-разному.

В CodeVisionAVR можно сделать глобальное определение (см. файл ft245\_functions.c)

```
#define _ALTERNATE_PUTCHAR_
```

и переопределить putchar(). После этого вывод всех функции, например, printf(), которые используют putchar(), будет перенаправлен в требуемое устройство, например, USART или USB-чип.

В avr-gcc используется стандартный для языка C подход. В файле nn\_usb\_link\_test.c определен новый поток

```
static FILE mystdout = FDEV_SETUP_STREAM(put_char, NULL, _FDEV_SETUP_WRITE)
```

и новая функция put\_char(), в которой и происходит выбор устройства вывода.

Необходимо отметить, что в библиотеке функций для FT245RL есть два варианта функций: с блокирующим и не блокирующим выводом. В случае блокирующего вывода

микроконтроллер будет ожидать от USB-чипа момента разрешения начала цикла отсылки данных.

## Программа проверки качества связи

Для тестирования радиосвязи по стандарту nanoNET (IEEE 802.15.4a) необходимо два модуля, условно обозначенных MASTER и SLAVE.

В начале цикла измерений MASTER в течение 1 секунды посылает узлу SLAVE кадры длиной 128 байт на максимальной выходной мощности. В журнал работы заносится распределение переданных кадров в зависимости от номера попытки (ретрансмиссии, attempts). В настройках трансиверов разрешено три попытки. Программа написана таким образом, что во многих случаях к числу неуспешных попыток будет добавлена 1, хотя на самом деле передача была успешной.

После этого узел MASTER последовательно посылает узлу SLAVE кадры, постепенно уменьшая амплитуду сигнала со значения 63 до 0, одновременно фиксируя количество ретрансмиссий. Если счетчик попыток передачи пакета для текущей мощности сигнала равнялся трем, это означало, что пакет так и не был доставлен адресату (узлу SLAVE). Пакеты подтверждения о приеме посылались всегда на максимальной мощности (63).

Типичная запись в журнале эксперимента (на экране) может выглядеть следующим образом.

**<RTC: 000000381CBF**

Packets/second distribution, 3 attempts - max:

```
1 attempt:      695
2 attempt:       0
3 attempt:       0
Unsuccess:      1
TOTAL          :   696
```

```
121 2333333333
      3333333333
2 1 1 333333333
```

876543221100999888777766666555555444444333333322222211111100000

**<RTC: 00000059C549**

Packets/second distribution, 3 attempts - max:

```
1 attempt:      348
2 attempt:       62
3 attempt:       43
Unsuccess:      75
TOTAL          :   528
```

```
1 33 33 333333
1 3333 3 3312 133
2333 21 333 2132
```

876543221100999888777766666555555444444333333322222211111100000

**RTC: 000000381CBF** – временная метка регистрации данных (аппаратная поддержка в трансиверах nanoNET).

Строчка, обозначенная синим цветом на рисунке 3, содержит набор цифр, каждая из которых обозначает уровень мощности отправленного информационного кадра. Всего 19 градаций – от 18 (написана только восьмерка, а единица для компактности в записи в журнале опущена) до 0.

Предыдущие три строки (для увеличения достоверности) соответствуют сериям отправки кадров с уменьшающейся силой сигнала.

Каждый символ в этих строчках обозначает количество ретрансмиссий, которое требуется для подтвержденной передачи. Если их не потребовалось, т.е. кадр был передан с первой

попытки, вставляется пробел. Если в серии посылки кадров с уменьшающейся амплитудой сигнала накапливается более 30 ретрансмиссий, серия обрывается.

Например (см. пример из журнала), в первой серии кадры с уровнем мощности 18, 17, 16 и т.д. до 3 включительно отсылались с первой попытки. А вот при уровне сигнала в 2 условные единицы, потребовались одна или две дополнительные ретрансмиссии; далее на первом уровне мощности вообще не было зарегистрировано безошибочных передач.

Другими словами, пока сигнал узла-мастера был достаточно сильным, узел SLAVE подтверждал прием каждого пакета. Как только уровень мощности стал равным 2, начали появляться проблемы с приемом. А для уровней сигнала от 1 до 0 вообще не было зарегистрировано ни одной успешной передачи.

То есть чем хуже были условия приема-передачи, тем ближе к началу третьей строки возникали цифры 1, 2 и 3 (см. вторую запись в журнале).

## Выбор конфигурации проекта (MASTER|SLAVE)

Выбор условий компиляции проекта для устройств MASTER или SLAVE проводится определением

```
#define MASTER
```

в файле nn\_usb\_link\_test.c. В случае выбора конфигурации SLAVE эту строчку необходимо закомментировать (`//#define MASTER`).

## Выбор конфигурации проекта (устройство вывода информации USB|USART)

Выбор устройства вывода проводится соответствующим присвоением значения переменной `output`:

```
//output=USART; // USART|USB
output=USB; // USART|USB
```

В случае выбора конфигурации MASTER + USB после подключения модуля к шине USB (включения питания радиомодуля) и старта терминальной программы (ее не запустить до момента подключения модуля, поскольку виртуальный COM-порт на тот момент еще не существует), программное обеспечение в ATmega32L (радиомодуль) будет ожидать нажатия любой клавиши на клавиатуре компьютера, и только после этого начнется очередной цикл измерения.

## Выбор режимов работы трансиверов

Также необходимо отметить, что выбор режимов работы обоих трансиверов лучше проводить при выключенном приемнике. Для этих целей служит блок в функции `main()`, ограниченный остановом и повторным запуском ресивера (отмечены синим).

```
NTRXSetRegister(TRX_RX_STOP, TRUE);
NTRXSetRegister(TRX_TX_ARQ_MAX, 0x02);
NTRXSetRegister(TRX_TX_ARQ, TRUE);
NTRXSetRegister(TRX_RX_ARQ_MODE, arqModeCrc2);
// NTRXSetRegister(TRX_TX_ARQ_MAX, 0x00);
// NTRXSetRegister(TRX_TX_ARQ, FALSE);
// NTRXSetRegister(TRX_RX_ARQ_MODE, arqModeNone);
temp=63; // power = MAX for Ack, Req2S and Clr2S
NTRXSetIndexReg(0);
```

```
transSPI(WRITE_CMD,SRfTxOutputPower1_O,&temp,1);  
transSPI(WRITE_CMD,SRfTxOutputPower0_O,&temp,1);  
NTRXSetRegister(TRX_MOD_ARRAY, mod4ary);  
// NTRXSetRegister(TRX_MOD_ARRAY, mod2ary);  
// NTRXSetRegister(TRX_RX_ADDR_MATCH, FALSE);  
NTRXSetRegister(TRX_RX_START, TRUE);
```

## Настройка RS-232

Настройка USART производится в функции InitPorts() (файл appl\_functions.c). По умолчанию выставляются следующие значения:

8 бит данных, 1 стоп-бит, без бита четности, 38400 бод.