СРЕДА BeeKit™ – УНИВЕРСАЛЬНЫЙ ИНСТРУМЕНТ РАЗРАБОТКИ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ БЕСПРОВОДНЫХ ПРИЛОЖЕНИЙ СТАНДАРТА 802.15.4/ZigBee. ЧАСТЬ 3*

Михаил Соколов, инженер, МЭИ (ТУ)
Александр Гришин, инженер, МЭИ (ТУ)

В этой части статьи, посвященной программному обеспечению Simple MAC (SMAC), входящему в состав среды проектирования BeeKit компании Freescale, рассказывается о структуре данных и о функциях ПО SMAC. Также кратко затронуты вопросы настройки проекта в среде BeeKit. Статья продолжает тему разработки программного обеспечения беспроводных устройств стандарта 802.15.4/ZigBee.

ТИПЫ, СТРУКТУРА ДАННЫХ ПО SMAC

В ПО SMAC приняты следующие типы данных [1]:

– UINT8 – беззнаковый 8-бит;
– UINT16 – беззнаковый 16-бит;
– UINT32 – беззнаковый 32-бит;
– INT8 – знаковый 8-бит;
– INT16 – знаковый 16-бит;
– INT32 – знаковый 32-бит.

Все перечисленные типы используются в проектах на базе SMAC и определяются в файле pub_def.h.

Рассмотрим более подробно структуру передаваемых и принимаемых массивов данных. Посредством структуры tTxPacket задаются данные для передачи по радио в ПО SMAC. Определение структуры находится в файле pub_def.h и выглядит следующим образом:

```
typedef struct {
UINT8 u8DataLength;
UINT8 *pu8Data;
} tTxPacket.
```

Составляющие структуры:

– u8DataLength – число байт, предназначенных для передачи;

– *pu8Data – указатель на массив передаваемых данных.

Методика использования структуры:

– в коде программы необходимо объявить переменную gsTxPacket типа tTxPacket;

– далее создается буфер данных для передачи: UINT8 gau8TxDataBuffer [20];

– затем указывается фактическое число байт, которое планируется передавать по радио (u8DataLength в tTxPacket-структуре): gsTxPacket.u8DataLength = 0;

– далее загружается адрес буфера данных в поле данных структуры gsTxPacket: gsTxPacket.pu8Data = = *gau8TxDataBuffer [0].

Структура tRxPacket определяет переменную, в которой будут помещаться данные, полученные по радио. Определяется структура в файле pub_def.h. Определение структуры выглядит следующим образом:

```
typedef struct {
UINT8 u8MaxDataLength;
UINT8 u8DataLength;
UINT8 *pu8Data;
UINT8 u8Status;
} tRxPacket.
```

Составляющие структуры:

– u8MaxDataLength – данная область определяет максимальный размер принимаемого пакета. Если принятый пакет превышает значение u8MaxDataLength, он отбрасывается, а приложение не прерывается;

– u8DataLength – область содержит длину полученного пакета в байтах;

– *pu8Data – указатель на массив, в который необходимо поместить принятые данные для последующей обработки основным приложением;

– u8Status – поле может принимать два возможных состояния: SUCCESS, TIMEOUT. Исходя из значения данного параметра, приложение может определить, данные были приняты успешно или нет. В функции MCPSDataIndication необходимо проверить эту область и убедиться, что прием данных завершился успешно.

Рассмотрим подробнее часть кода по инициализации МК, радиомодема и периферии. Эта часть кода расположена в начале основной функции программы, встречается практически во всех проектах и имеет следующий вид:

```
MCUInit(); //Инициализация МК;
MCUPortInit(); //Настройка неиспользуемых портов для минимизации энергопотребления;
MC13192Init(); // Инициализация радиомодема;
LEDInit(); /* Конфигурирование портов МК, задействованных для управления светодиодами и управляющими переключателями */;
MLMESetMC13192ClockRate(0); /* 16 MHz MC13192 CLKO – активация частоты тактирования от радиомодема */;
UseExternalClock(); /* Настройка частоты внутренней шины МК на 8 Гц и перевод МК на тактирование от радиомодема */;
SPMSC2 = SPMSC2&~0x03; /* Активация режима пониженного энергопотребления МК-stop3. */;
TPM1SC = 0x0F; /* Задание предделителя модуля таймера – 128. */;
MC13192_IRQ_IE_BIT = 1; /* Активация прерываний от радиомодема */;
EnableInterrupts; // Разрешение прерываний в МК
MLMESetChannelRequest(Chan_Num); // Задание номера рабочего радиоканала;
MLMESetMC13192TmrPrescale(3); /* Настройка внутреннего таймера радиомодема – 250 кГц */;
```

* Часть 1 опубликована в журнале «Электронные компоненты» №3/2007. Часть 2 опубликована в журнале «Электронные компоненты», 2007, №5.

NEW!

NEC

32-разрядные микроконтроллеры

для теплосчетчиков, газоанализаторов, весов и сложных электросчетчиков

ЯДРО V850

- 32/16-разрядная RISC архитектура (29 MIPS при 20 МГц)
- Малые электромагнитные шумы
- Малое энергопотребление
- DSP команды

- FLASH: от 128 до 640 КБ
- ОЗУ: от 12 до 48 КБ
- Pin: от 100 до 144
- Бесплатный Си-компилятор (128 КБ)

Техническое сопровождение проектов

Элтех

Официальный дистрибьютор

Оценочный комплект 56 € со склада компании Элтех

www.eltech.spb.ru
nec@eltech.spb.ru

Санкт-Петербург
ул. Победы, 11
тел. (812) 327-9090

Москва
ул. Угрешская, 2, стр.1
тел. (495) 788-5948

Екатеринбург
тел. (343) 377-7094
(343) 257-7037

Ростов-на-Дону
тел. (863) 220-3071
(863) 220-3072

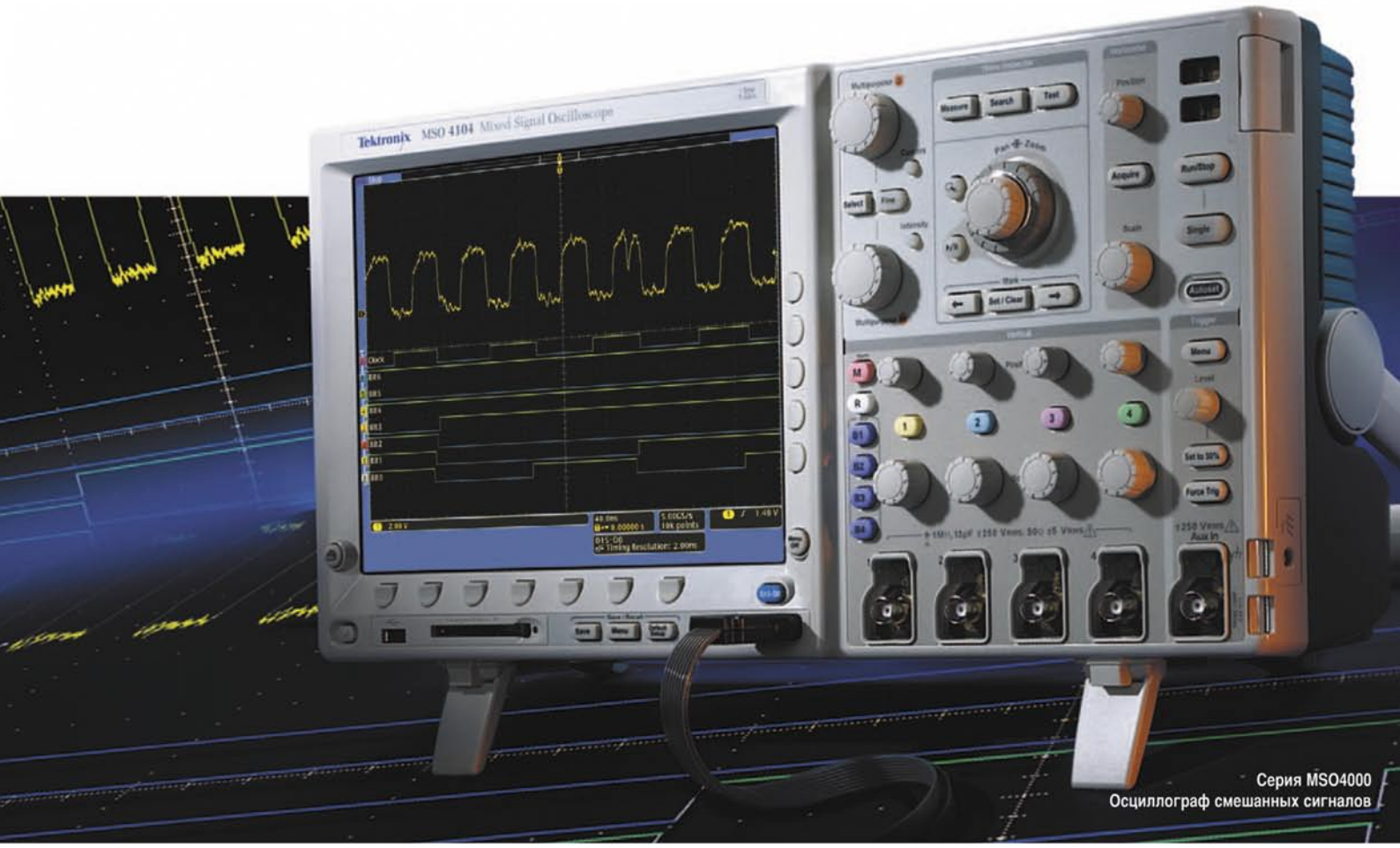
Новосибирск
тел. (383) 212-5874
(383) 212-5875

Ижевск
тел. (3412) 600-660
(3412) 600-661

Представительство в Минске
БЕЛЭЛТЕХ
тел. (375 17) 256-1860
(375 17) 256-1861

Множество сигналов, один прибор

Высокопроизводительный осциллограф с возможностями логического анализатора



Серия MSO4000
Осциллограф смешанных сигналов

Теперь с помощью одного прибора Tektronix можно выполнять визуализацию и корреляцию как аналоговых, так и цифровых сигналов. Его просто настраивать и им легко пользоваться. Представляем Вам осциллограф смешанных сигналов серии MSO4000. Это полнофункциональный осциллограф с основными возможностями, присущими логическому анализатору. Его дисплей с разнообразными возможностями настройки для осциллограмм с 2 или 4 аналоговыми и 16 цифровыми каналами позволяет одновременно просматривать множество сигналов. Все операции с осциллограммами хорошо знакомы, поэтому отладка схем будет проходить быстрее и проще, чем раньше. Упрощена процедура настройки и применения. У Вас появится больше времени на проведение исследований.

Рабочие характеристики

Модели	MSO4032, MSO4034, MSO4054, MSO4104
Полоса пропускания	350 МГц, 500 МГц, 1 ГГц
Каналы	2 или 4 аналоговых + 16 цифровых
Частота выборки в аналоговых каналах	До 5 Гвыб/с
Макс. временное разрешение в цифровых каналах	60,6 пс
Длина записи	10 млн. точек во всех аналоговых и цифровых каналах
Дисплей	Большой XGA-экран размером по диагонали 10,4 дюйма (264 мм)

SERNIA

Тел.: (495) 225 40 14, Факс: (495) 932 92 44
www.sernia.ru, se@sernia.ru

Tektronix®
Enabling Innovation

MLMEMC13192PAOutputAdjust (MAX_POWER); /* Задание выходной мощности радиопередатчика */.

В приведенном примере по инициализации красным цветом выделены функции, входящие в состав ПО SMAC.

ФУНКЦИИ, ВХОДЯЩИЕ В SMAC, ПРОТОКОЛ ПЕРЕДАЧИ

ПО SMAC является упрощенной версией ПО MAC, реализующего весь функционал стандарта IEEE802.15.4. Соответственно, возможности SMA меньше, а сами функции достаточно простые, они требуют минимум флэш-памяти и предоставляются разработчику в исходных кодах. Перечень функций представлен ниже.

MCPSDataRequest – функция для передачи пакета данных. После успешной передачи пакета возвращает результат {SUCCESS}.

MCPSDataIndication – вызываемая функция, необходима для работы SMAC, уведомляет приложение о приеме пакета данных. Помещается внутрь основной программы пользователя, что позволяет ПО SMAC вызывать ее при приеме данных по радио. Возвращает: результат {SUCCESS} – успешно принятый пакет, {TIMEOUT} – произошел таймаут в результате нахождения радиомодема в режиме приема в течение определенного времени, {OVERFLOW} – произошло переполнение при приеме числа байт, превышающего максимальную длину пакета.

MLMEHibernateRequest – функция переводит трансивер в энергосберегающий режим Hibernate. При этом частота на выводе модема Clock Output (CLKO) выключается. В случае, если МК тактировался от радиомодема, пользователь должен сначала перевести МК на внутреннее тактирование и только потом вызвать функцию MLMEHibernateRequest(). Результат вызова функции – {SUCCESS}. Примечание: пользователь может вызвать функцию MLMEWakeRequest (), чтобы вывести модем из состояния Hibernate. После перевода радиомодема в нормальный режим работы можно снова переключать МК на внешний источник тактирования.

MLMEWakeRequest – функция выводит трансивер из режимов энергосбережения Hibernate или Doze. Результат: {SUCCESS} – в случае вывода модема из одного из указанных режимов, {ERROR} – если во время вызова функции радиомодем находился в любом другом режиме либо не функционировал.

MLMESetChannelRequest – с помощью данной функции выбирается рабочий частотный канал, на котором будут происходить передача и прием пакетов данных в радиоэфире. Результат: {SUCCESS} – если в качестве аргумента задано число из диапазона 0–15 (в десятичном виде), в противном случае – {ERROR}.

MLMERXEnableRequest – функция переводит трансивер в режим приема данных на радиоканале, заданном с помощью функции MLMESetChannelRequest. При этом создается буфер для записи и хранения поступающих данных, которые могут быть прочитаны пользователем после вызова функции MCPSDataIndication. Приложение: будет находиться в режиме приема, пока не будет вызвана функция MLMERXDisableRequest, пока значение переменной u32Timeout равно нулю. Результат: {SUCCESS} – успешный прием пакета данных, {ERROR} – радиомодем не перешел в режим приема.

MLMERXDisableRequest – функция возвращает трансивер в состояние Idle из состояния приема, возвращает

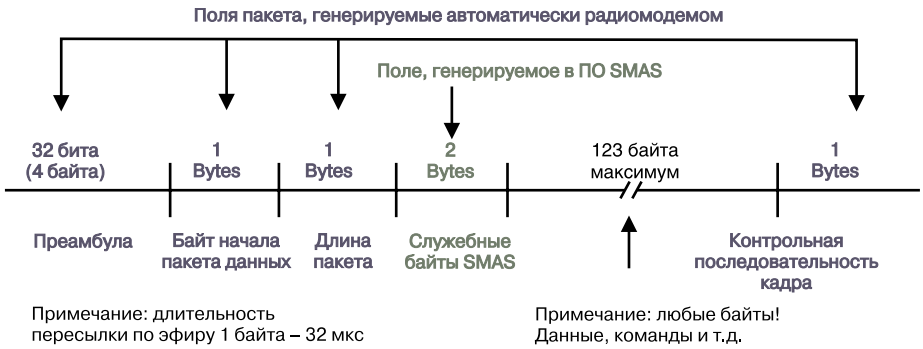


Рис. 1. **Формат пакета передаваемых по эфиру данных SMAC**

результат {SUCCESS}. Может быть использована для принудительного выключения режима приема.

MLMESetMC13192ClockRate – функция задания и выбора частоты тактирования МК от радиомодема. Частота доступна на выводе модема CLKO, возвращает результат {SUCCESS}. Возможные варианты частот в зависимости от заданного значения представлены ниже:

– 0...16 МГц (значение по умолчанию);
– 1...8 МГц;
– 2...4 МГц;
– 3...2 МГц;
– 4...1 МГц;
– 5...62,5 КГц;
– 6...32,786 КГц;
– 7...16,393 КГц.

MLMEEnergyDetect – функция активирует режим детектирования уровня сигнала и определения занятости радиоэфира (Eenergy Detect/Clear Channel Assessment (CCA)), возвращает измеренное значение мощности в канале. Реальное значение мощности в dBm расчитывается по формуле power/2.

MLMEMC13192SoftReset – функция вызывает программный сброс радиомодема, возвращает результат {SUCCESS}. После программного сброса модема автоматически вызывается функция MLMEMC13192Reset Indication. Таким образом пользователь может получить подтверждение о выполнении операции сброса.

MLMEMC13192XtalAdjust – функция для подстройки внутренней частоты работы радиомодема, возвращает результат {SUCCESS}.

MLMELinkQuality – функция возвращает значение, определяющее мощность сигнала последнего принятого пакета данных, вычисляется по формуле: dBm = (–Link Quality/2), вызывается после функции MCPSDataIndication. Значение LinkQuality остается неизменным до принятия следующего пакета данных.

MLMESetMC13192TmrPresca e – посредством данной функции задается частота работы внутренних таймеров радиомодема, возвращает результат {SUCCESS}. Доступные варианты частоты работы таймеров в зависимости от аргумента функции:

– 0...2 МГц;
– 1...1 МГц;
– 2...500 кГц;
– 3...250 кГц (значение по умолчанию);
– 4...125 кГц;
– 5...62,5 кГц;
– 6...31,25 кГц;
– 7...15,625 кГц.

MLMEMC13192FEGainAdjust – используется для компенсации значения качества принимаемого сигнала LinkQuality в случае его переполнения, например при использовании внешнего входного малошумящего усилителя, возвращает результат {SUCCESS}.

MLMEDozeRequest – функция переводит радиомодем в режимы Normal Doze Mode (без частоты тактирования на выводе CLKO, но с автоматическим пробуждением) и Acoma Mode (с частотой тактирования на выводе CLKO, но без таймаутов пробуждения), возвращает результат {SUCCESS}. Для активации режима Acoma Mode необходимо вызвать функцию со значением аргумента «0». Для выхода из режима вызывается функция MLMEWakeRequest. Режим Normal Doze Mode активируется вызовом функции со значением аргумента, равным величине таймаута, при наступлении которого модем генерирует прерывание IRQ для МК. После этого радиомодем выдает еще 128 тактов для того, чтобы МК успел перейти в режим пониженного энергопотребления Wait. Необходимо иметь в виду, что частота тактирования на выводе CLKO в режиме Acoma Mode не превышает 1 МГц.

MLMEMC13192PAOutputAdjust – функция задает выходную мощность радиомодема, вызывается со значением аргументов от 0 до 15 в десятичном формате, либо с использованием макросов MIN_POWER, MAX_POWER и NOMINAL_POWER. Таблица соответствия значений аргумента и выходной мощности приведена в описании модема. Возвращает результат {SUCCESS} в случае успешной операции, {OVERFLOW} – при вызове функции со значением аргумента больше 15.

MLMEGetRficVersion – функция возвращает номер версии радиомодема.

MLMETestMode – функция по тестированию различных режимов работы радиомодема, перечень которых уже приводился в начале статьи, результат – void.

MLMEMC13192ResetIndication – функция помещается в коде основной программы пользователя и автоматически вызывается после сброса радиомодема, вызванного либо программным способом (с помощью функции MLMEMC13192SoftReset), либо в результате аппаратного сброса модема (подачей логического нуля на вывод RST-модема), результат – void.

MCUInit – функция выполняет инициализацию части МК, задействованной при работе с радиомодемом. Всегда вызывается перед инициализацией самого модема.

UseExternalClock – функция настраивает и переводит внутренний модуль генератора МК на работу от внешнего источника тактирования, в нашем случае на сигнал тактирования от радиомодема. Данная функция вызывается вслед за функцией MLMESetMC13192ClockRate.

UseMCUClock – функция настраивает и переводит МК на работу от внутреннего источника тактирования (внутренний генератор либо собственный кварцевый резонатор МК). Функция обычно используется перед переводом модема в режим пониженного энергопотребления.

MLMEScanRequest – функция сканирует определенные радиоканалы, используя один из двух возможных методов, и возвращает значение мощности на всех каналах.

MC13192DisableInterrupts – функция запрещает прерывания, генерируемые в ПО SMAC. Следует иметь в виду, что запрет всех прерываний от ПО SMAC при вызове некоторых

функций SMAC может привести к зависанию программы.

MC13192EnableInterrupts – функция разрешает прерывания, генерируемые в ПО SMAC.

MC13192ContReset – функция подает и удерживает логический ноль на линии сброса радиомодема RST, при этом модем переходит в режим пониженного энергопотребления OFF (ток потребления составляет менее 1 мкА, обычно 0,2 мкА).

MC13192Restart – функция возвращает логическую единицу на линию сброса радиомодема RST, тем самым переводя его из состояния OFF в активный режим.

Протокол передачи данных по радиоканалу представлен на рисунке 1.

НАСТРОЙКА ПРОЕКТА В СРЕДЕ BEEKIT, ЭКСПОРТИРОВАНИЕ В CODEWARRIOR IDE

Разработка собственного ПО для беспроводных приложений 802.15.4/ZigBee включает в себя два этапа, вне зависимости от используемой библиотеки – работа со средой Beekit и программирование в среде CodeWarrior IDE.

Рассмотрим первый этап – создание и настройка проекта в BeeKit. Функциональные возможности среды BeeKit кардинально отличаются при использовании библиотек SMAC, MAC или BeeStack. Если разработчик выбрал библиотеку MAC или BeeStack, в среде можно настраивать и задавать значения практически всех параметров, заложенных в стандарт 802.15.4 и технологию ZigBee. Библиотека SMAC благодаря своей простоте подразумевает настройку лишь нескольких из них:

– **Security Enabled** – подключает/отключает в проекте модуль безопасности для автоматического шифрования передаваемых по радио пакетов;

– **OTAP Enabled** – подключает/отключает в проекте модуль перепрограммирования FLASH по радио;

– **Promiscuous Mode** – при активации параметра два служебных байта SMAC не включаются в протокол передачи пакетов;

– **OTAP Default ID** – задает идентификатор по умолчанию для функции перепрограммирования по радио;

– **Security Type** – выбирает алгоритм шифрования в модуле безопасности (на данный момент доступен только один алгоритм – SIMPLE_XOR);

– **Target hardware** – позволяет выбрать тип целевой платы, на которой будет исполняться создаваемый программный код;

– **LCD Enabled** – подключает в проект поддержку ЖКИ;

– **Default SCI Port** – активирует и определяет последовательные UART интерфейсы МК;

– **Default Channel** – задает радиоканал по умолчанию в проекте;

– **Output Power** – задает значение выходной мощности по умолчанию в радиопередатчике.

После настройки параметров необходимо выполнить проверку и подтвердить правильность проекта и всех изменений, выбрав Validate Solution во всплывающем меню, возникающем после нажатия правой кнопки мыши на названии проекта в проводнике решений [2]. При отсутствии ошибок проект необходимо экспортировать в среду CodeWarrior IDE, выбрав Export Solution в том же всплывающем меню. Далее работа с проектом продолжается уже в среде CodeWarrior IDE, чему также будет посвящена одна из последующих публикаций.

ЛИТЕРАТУРА

1. www.freescale.com/zigbee.