



Всеволод Нестеров (КОМПЭЛ)

ИСПОЛЬЗОВАНИЕ РЕЖИМОВ ПОНИЖЕННОГО ЭНЕРГОПОТРЕБЛЕНИЯ РЧ-ТРАНСИВЕРОВ

Многообразие режимов работы с пониженным энергопотреблением позволяет использовать современные радиочастотные трансиверы в устройствах с батарейным питанием. В статье рассказывается о режимах пониженного энергопотребления трансиверов Texas Instruments CC111xFx/CC251xFx/CC243x из производственной линейки Chipcon.

Трансиверы CC111xFx/CC251xFx/CC243x имеют пять основных режимов работы:

Активный режим: Режим полной функциональности. Включен внутренний стабилизатор напряжения и питание подается на ядро. Активны высокочастотные тактовые генераторы — либо по отдельности, либо оба вместе. Также работают либо RC-, либо кварцевый генератор 32,768 кГц, либо они оба одновременно.

Режим PM0: Аналогичен активному, но процессор находится в ждущем режиме, то есть код не выполняется.

Режим PM1: Включен внутренний стабилизатор напряжения и питание подается на ядро. Ни один из тактовых генераторов не работает. Работает либо RC-, либо кварцевый генератор 32,768 кГц. Система перейдет в активный режим по сигналу RESET, или по сигналу внешнего прерывания, или по срабатыванию таймера спящего режима.

Режим PM2: Питание на ядро не подается. Ни один из тактовых генераторов не работает. Работает либо RC-, либо кварцевый генератор 32,768 кГц. Система перейдет в активный режим по сигналу RESET, или по сигналу внешнего прерывания, или по срабатыванию таймера спящего режима. Информация о состоянии USB для тран-

сивера CC2511Fx будет потеряна при вхождении в режим PM2. Таким образом, не следует использовать режим PM2 с USB.

Режим PM3: Питание на ядро не подается. Ни один из генераторов не работает. Система перейдет в активный режим либо по сигналу RESET, либо по сигналу внешнего прерывания. Информация о состоянии USB для трансивера CC2511Fx будет потеряна при вхождении в режим PM3. Этот режим также не следует использовать с USB.

Требуемый режим выбирается установкой битов SLEEP.MODE. После установки этих битов необходимо установить биты IDLE в регистре PCON SFR, что переведет трансивер в выбранный режим. Надо заметить, что при SLEEP.MODE≠0 прерывания и переключение между генераторами заблокированы. Также надо соблюдать гарантированное время нахождения тактовых генераторов в режимах пониженного энергопотребления. Если это время будет меньше, чем указано в описаниях, то могут быть коллизии.

На рисунке 1 указан пример кода на языке C для перевода трансиверов в режимы PM0-PM3.

Для трансиверов CC111xFx/CC251xFx при «пробуждении» из режимов PM{2-3} есть небольшая вероятность того, что SLEEP.



На рынке появился новый РЧ-модуль на базе CC1100

Радиочастотный модуль Panasonic PAN3550 выполнен на базе многоканального приемопередатчика (трансивера) CC1100 компании Texas Instruments (Chipcon) и микроконтроллера MC9S08GT60 компании Freescale. Радиочастотный модуль предназначен для применения в системах малого радиуса действия с двухсторонней передачей данных в безлицензионном диапазоне частот 868 МГц. Модуль PAN3550 идеально подходит для использования в качестве радиомодема промышленного стандарта RF Konnex™. Особенностью модуля является интегрированная антенна, выполненная в виде проводников печатной платы. Трансивер CC1100 позволяет гибко выбирать длину пакета и самостоятельно осуществляет полную обработку данных — генерацию преамбулы, вставку и обнаружение синхрослова, проверку адреса, автоматическое вычисление контрольной суммы CRC. Модуль может использоваться как автономный законченный узел, благодаря возможности выполнения прикладной программы пользователя в микроконтроллере MC9S08GT60.

Особенности PAN3550:

- Законченный радиомодем,
- Встроенная антенна,
- Интерфейсы GPIO(14), UART (2), I²C, A/D (4),
- 32 Кб Flash-памяти программ, 2 Кб RAM-памяти,
- Оценка занятости канала и качества соединения,
- Аппаратная коррекция ошибок.

Основные технические характеристики:

- Высокая чувствительность: -110 дБм при 1,2 кбит/с,
- Программируемая скорость передачи данных: от 1,2 до 500 кбит/с,
- Выходная мощность до +6 дБм (программируемая),
- Температурный диапазон от -40 до 85°C,
- Размеры: 20,3 x 26 x 3 мм.

```

/* код на языке C: */
void main(void)
{
/* Здесь необходимо вставить соответствующий код, указанный на
/* рисунках 6 и 7. Не подходит для режима PM0! */

/* Настройка и разрешение прерываний (Sleep-таймер, порт)
/* по которым трансивер будет «просыпаться» */

/* Замечание:
/* Для входа в режим PM{2 - 3} для CC111xFx/CC251xFx следующий
/* блок кода должен быть заменен кодом из рисунка 2 */

/* Начало блока кода */

/* Установка SLEEP.MODE в соответствии с нужным режимом PM,
/* например, PM1. */
SLEEP = (SLEEP & 0xFC) | 0x01;

/* Необходимо вставить три NOP для того, чтобы позволить
/* процессу блокирования прерываний закончиться перед следующей
/* проверкой битов SLEEP.MODE. Надо заметить, что прерывания
/* блокируются при SLEEP.MODE ≠ 0, таким образом, время между
/* установкой SLEEP.MODE ≠ 0, и записью в PCON.IDLE должно быть
/* как можно меньше. Если прерывание возникнет перед тем, как
/* команды NOP закончатся, процедура обработки прерываний
/* очистит SLEEP.MODE-биты в соответствии с кодом на рисунке 8. */

asm("NOP");
asm("NOP");
asm("NOP");

/* Если во время команд NOP прерывание не поступит,
/* то прерывания эффективно заблокируются в этом месте кода
/* Если биты SLEEP.MODE будут очищены в этом месте, это будет
/* означать, что их очистила процедура обработки прерываний
/* во время команд NOP и трансивер не войдет в режим PM. */

if (SLEEP & 0x03)
{
/* Установка PCON.IDLE для входа в выбранный режим PM, т.е. PM1. */
PCON |= 0x01;

/* Трансивер сейчас в выбранном режиме и «проснется»
/* по прерыванию таймера или внешнего порта */

/* Первая инструкция после выхода из режима PM. */
asm("NOP");

}
/* Конец блока кода */

/* Здесь надо вставить соответствующий код, указанный
/* на рисунках 6 и 7. Не подходит для режима PM0! */
...
}

```

Рис. 1. Пример кода на языке C для перевода трансиверов в режимы PM0...PM3

MODE-биты будут ошибочно установлены в отличное от нуля значение перед тем, как бит PCON.IDLE будет очищен микроконтроллером. Вследствие этого трансиверы CC111xFx/CC251xFx немедленно вновь перейдут в режим PM{2-3}. Так как причина пробуждения (прерывание) все еще будет активна в этот момент, трансиверы CC111xFx/CC251xFx «проснутся» и снова перейдут в режим PM{2-3}, что приведет к зависи-

нию, из которого можно выйти только по сигналу RESET.

Для того чтобы быть уверенным, что биты SLEEP.MODE установлены в 0, рекомендуется передавать посредством DMA-передачи определенное количество байт в регистр SLEEP. Этот регистр переключается записью DMAREQ.DMAREQx битов непосредственно перед записью бита PCON.IDLE, однако это требует определенных условий:

```

/* код на языке C: */

/* Инициализация буферов источников и DMA дескрипторов для DMA
/* передачи. */
unsigned char __xdata PM2_BUF[7] =
{0x06, 0x06, 0x06, 0x06, 0x06, 0x06, 0x04};
unsigned char __xdata PM3_BUF[7] =
{0x07, 0x07, 0x07, 0x07, 0x07, 0x07, 0x04};
unsigned char __xdata dmaDesc[8] =
{0x00, 0x00, 0xDF, 0xBE, 0x00, 0x07, 0x20, 0x42};

void main(void)
{
...

/* сохранение текущего DMA дескриптора нулевого канала 0
/* и остановка текущих передач, если канал используется
/*
unsigned char storedDescHigh = DMA0CFGH;
unsigned char storedDescLow = DMA0CFGH;
DMAARM |= 0x81;

/* Связывание дескриптора с правильным источником
/*
/* Замените &PM2_BUF на &PM3_BUF если используется PM3 */

dmaDesc[0] = (unsigned int)& PM2_BUF >> 8;
dmaDesc[1] = (unsigned int)& PM2_BUF;

/* Связывание дескриптора с DMA каналом 0 и его активация
/*
DMA0CFGH = (unsigned int)&dmaDesc >> 8;
DMA0CFGH = (unsigned int)&dmaDesc;
DMAARM = 0x01;

/* Замечание! В этом месте надо убедиться, что все прерывания,
/* которые не будут использоваться для «пробуждения»
/* трансивера, были отключены, как описано в главе "Power
/* Management Control" описания.

/* Синхронизация с положительным фронтом сигнала генератора
/*
/* 32 кГц как описано в главе "Sleep Timer and Power Modes".
/*
char temp = WORTIME0;
while (temp == WORTIME0);

/* Убедитесь, что генератор XOSC отключен при входе в режим
/*
/* PM{2 - 3} и что кэш отключен.
/*
/* Замените 0x06 на 0x07, если используется Pm3
/*
MEMCTR |= 0x02;
SLEEP = 0x06;

/* Вход в режим так, как рассказано в главе "Power Management
/* Control" описания. Убедитесь, что DMA-канал 0 активирован
/* непосредственно перед установкой PCON.IDLE.
/*
asm("NOP");
asm("NOP");
asm("NOP");
if (SLEEP & 0x03)
{
asm("MOV 0xD7, #0x01"); /* DMAREQ = 0x01;
asm("NOP"); /* Needed to perfectly align the
/* the DMA transfer.
asm("ORL 0x87, #0x01"); /* PCON |= 0x01;
asm("NOP");
}

/* Включение кэша.
MEMCTR &= ~0x02;

/* Связывание DMA канала 0 с оригинальным дескриптором и его
/* инициализация, если он использовался перед входом в режим PM
/*
DMA0CFGH = storedDescHigh;
DMA0CFGH = storedDescLow;
DMAARM = 0x01;

...
}

```

Рис. 2. Пример кода для начала DMA-передачи

- Трансивер CC111xFx/CC251xFx должен работать на максимально возможной частоте HS RC-генератора.

- Генератор HS XOSC должен быть отключен

- Кэш-память должна быть отключена

Следующий код на рисунке 2 предполагает, что трансивер CC111xFx/CC251xFx уже работает на максимально возможной скорости HS RC-генератора. Код,

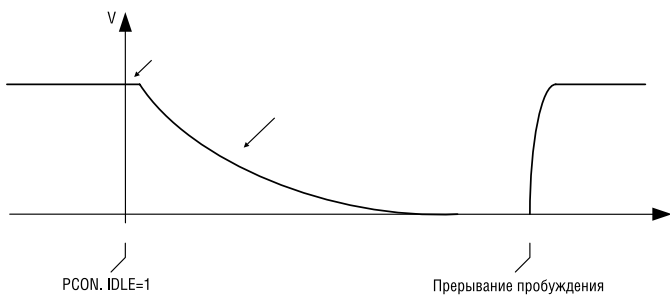


Рис. 3. Напряжение на регуляторе в режиме PM{2 - 3}

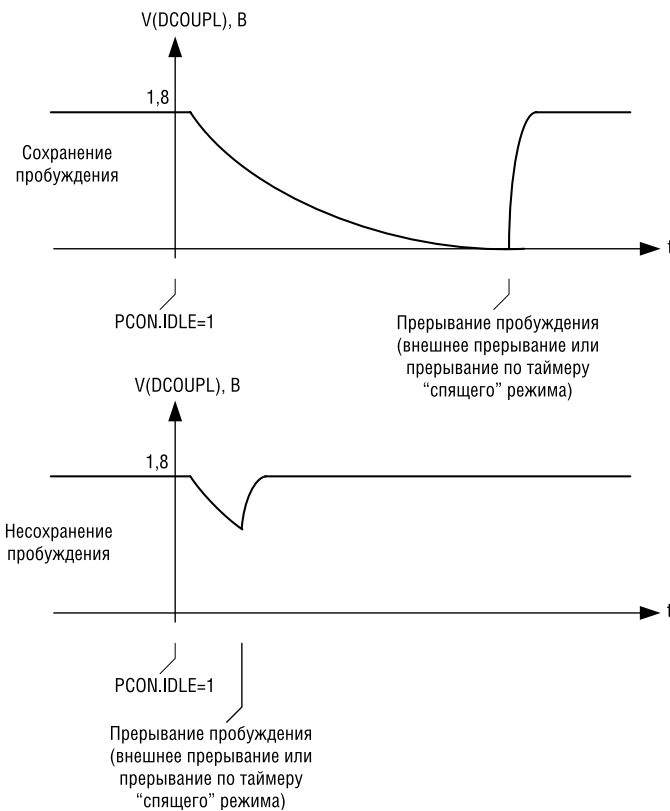


Рис. 4. Гарантированные и негарантированные «пробуждения» трансивера CC2430

выделенный жирным цветом, критичен ко времени выполнения, поэтому должен быть использован без изменений.

Трансивер CC243x не имеет таких проблем с пробуждением, как CC111xFx/CC251xFx. Таким образом, процедура ввода CC243x в режим PM{2-3} такая же, как описана на рисунке 1. Надо лишь заметить, что CC243x должен вводиться в режим PM{2-3} на максимально возможной скорости HS RC-генератора. Однако в кристаллах версии D для CC243x есть

другая проблема с «пробуждением» из режимов PM{2-3}.

Трансивер CC243x имеет встроенный регулятор, который выдает напряжение 1,8 В для питания ядра при входных 2,0...3,6 В. Типичное напря-

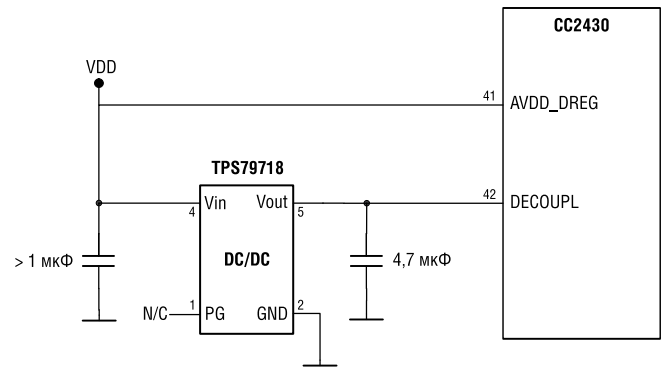


Рис. 5. Принципиальная схема применения внешнего регулятора напряжения

```

/* код на языке C: */
Void main(void)
{
...
/* Следующий код должен быть выполнен как перед входением */
/* в режим PM, так и после «пробуждения»: */
/* Очистка CLKCON.OSC для того, чтобы трансивер */
/* CC111xFx/CC251xFx работал на частоте генератора HS XOSC. */
/* Это необходимо, если генератор HS XOSC не был запущен перед */
/* входом в режим PM */
CLKCON &= 0xBF;

/* Проверка SLEEP.XOSC_STB для того, чтобы быть уверенным, что */
/* генератор HS XOSC стабильно работает перед продолжением */
/* выполнения кода */
while(!(SLEEP & 0x40));

/* Установка SLEEP.OSC_PD для выключения генератора HS RCOSC */
SLEEP |= 0x04;
...
}

```

Рис. 6. Проверка/инициализация генераторов для CC111xFx/CC251xFx

```

/* код на языке C: */
Void main(void)
{
...
/* Следующий код должен быть выполнен как перед входением в */
/* режим PM, так и после «пробуждения». Однако, надо заметить, */
/* что он необходим, только если необходимо запустить генератор */
/* HS XOSC в активном режиме: */
/* Установка SLEEP.OSC_PD для включения генератора HS XOSC */
/* (генератор HS RCOSC уже запущен) */
SLEEP &= 0xFFB;

/* Проверка SLEEP.XOSC_STB для того, чтобы быть уверенным, что */
/* генератор HS XOSC стабильно работает перед продолжением */
/* выполнения кода */
while(!(SLEEP & 0x40));

/* В соответствии с описанием CC243x необходимо добавить */
/* соответствующее количество команд NOP, что дает 64 мкс */
/* задержки перед включением генератора HS XOSC */
asm("NOP");

/* Очистка CLKCON.OSC для включения генератора HS XOSC. */
CLKCON &= 0xBF;

/* Проверка CLKCON.OSC, для того, чтобы быть уверенным, что */
/* трансивер CC243x переключился на работу от генератора */
/* HS XOSC */
while (CLKCON & 0x40);

/* Установка SLEEP.OSC_PD для выключения генератора HS RCOSC */
SLEEP |= 0x04;
...
}

```

Рис. 7. Проверка/инициализация генераторов для CC243x

```

/* код на языке C: */
/* процедура обработки прерываний таймера спящего режима (ISR) */
#pragma(vector=VECT(5, 0x2B)) __near_func __interrupt void
ST_ISR(void);

#pragma(vector=VECT(5, 0x2B)) __near_func __interrupt void
ST_ISR(void)
{
/* Очистка IRCON.STIF (флаг прерывания таймера спящего режима) */
IRCON &= 0x7F;

/* Очистка WORIRQ.EVENT0_FLAG (флаг внешнего прерывания таймера
спящего режима) */
/* Это необходимо только для трансивера CC111xFx/CC251xFx! */
WORIRQ &= 0xFE;

...

/* Необходимо очистить биты SLEEP.MODE для того, чтобы
прерывание также могло возникнуть и перед тем как трансивер
войдет в режим PM. Если это прерывание возникнет в течении
работы трех команд NOP (перед тем, как прерывания
заблокируются) в коде из рисунка 1, то очистка битов
SLEEP.MODE будет гарантировать, что приложение не войдет
в режим PM.
SLEEP &= 0xFC;
}

/* Процедура обработки внешних прерываний, например из порта 1 */
#pragma(vector=VECT(15, 0x7B)) __near_func __interrupt void
Pl_ISR(void);

#pragma(vector=VECT(15, 0x7B)) __near_func __interrupt void
Pl_ISR(void)
{
/* Очистка P1IFG.bit1 (флага внешнего прерывания Port1.Pin1) */
P1IFG = 0xFD;

/* Очистка IRCON2.P1IF (флага прерываний порта 1) */
IRCON2 &= 0xF7;

...

/* Необходимо очистить биты SLEEP.MODE для того, чтобы
прерывание также могло возникнуть и перед тем как трансивер
войдет в режим PM. Если это прерывание возникнет в течении
работы трех команд NOP (перед тем, как прерывания
заблокируются) в коде из рисунка 1, то очистка битов
SLEEP.MODE будет гарантировать, что приложение не войдет
в режим PM.
SLEEP &= 0xFC;
}

```

Рис. 8. Код возврата в активный режим

Таблица 1. Потребляемые токи в каждом из режимов

Интервал	Описание	Ток, мА	Длительность, мс	Потребление, мА/мс
1	Переход из спящего режима в режим генератора HS RCOSC 16 МГц	0,75	0,49	0,3675
2	Переход в режим генератора HS XOOSC 32 МГц	12	1,8	21,6
3	CMSA/CA алгоритм. Трансивер в режиме приема.	31	1,6	32,86
4	Переключение из режима приема в режим передачи	18	0,19	3,42
5	Передача пакетов. Трансивер в режиме передачи.	29	0,58	16,82
6	Переключение из режима передачи в режим приема	18	0,11	1,98
7	Прием подтверждения от координатора. Трансивер в режиме приема	31	1,1	34,1
8	Обработка пакетов. Ядро в режиме 32 МГц	12	1,1	13,2
9	Переход в спящий режим (PM0 → PM2). Ядро в режиме 16 МГц.	7	0,62	4,34

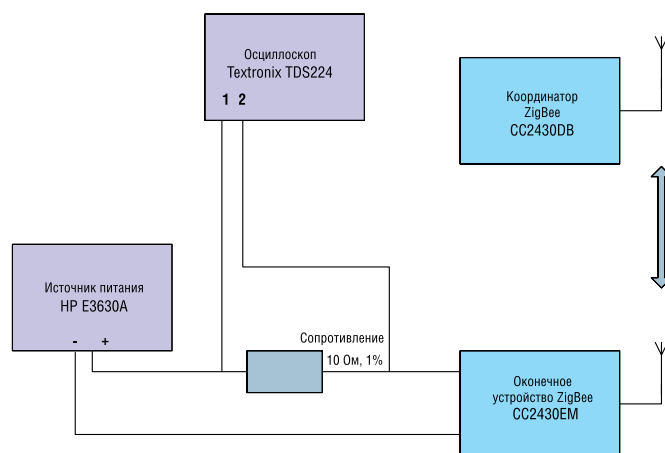


Рис. 9. Блок-схема изменения потребления тока

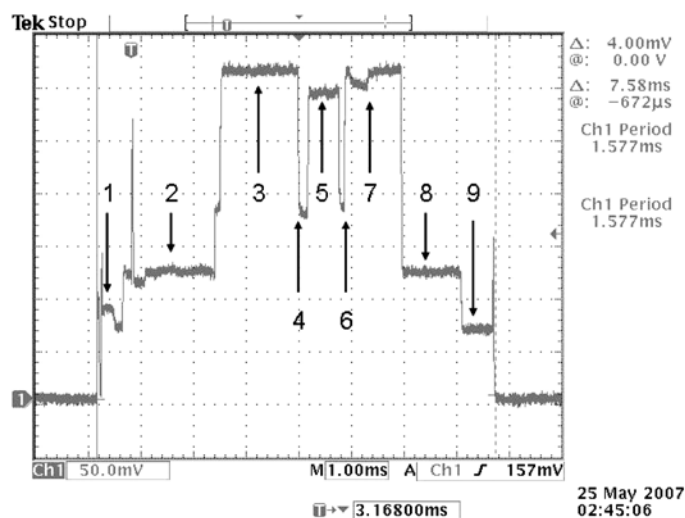


Рис. 10. Вид измерений на экране осциллографа

жение в режиме PM{2-3} показано на рисунке 3.

Проблема заключается в том, что если прерывание на «пробуждение» поступает в момент, когда напряжение на выходе регулятора находится в пределах 0,5...1,6 В, то это может привести к зависанию ядра, потерям данных в оперативной памяти и регистрах и другим проблемам (рис. 4).

Для борьбы с этим явлением рекомендуется использование внешнего регулятора (рис. 5).

При возвращении трансиверов CC111xFx/CC251xFx (рис. 6) и CC243x (рис. 7) в активный режим важно правильно сделать проверку/инициализацию генераторов.

Собственно код возврата в активный режим для трансиверов CC111xFx/CC251xFx/CC243x: показан на рисунке 8.

При использовании для питания трансиверов аккумуляторов важно знать, за какое время устройство разрядит используемый элемент. Для определения потребления можно использовать следующую схему (рис. 9) на примере ZigBee-трансиверов CC2430.

На экране осциллографа измерения могут выглядеть примерно так, как показано на рисунке 10.

Итого имеем следующие данные, обобщенные в таблице 1.

Вычисляем общее потребление, оно получается 128,7 мА/мс.

Попробуем рассчитать время жизни батареи при односекундных интервалах опроса:

Потребление во время опроса:

$$\frac{128,7 \text{ мА/мс} \times 60 \text{ с/мин} \times 60 \text{ мин/ч} \times 24 \text{ ч/сут}}{1000 \text{ мс/с} \times 3600 \text{ с/ч}} = 3,088 \text{ мА/ч}$$

Потребление во время сна:

$$\frac{0,0005 \text{ мА} \times (1000 \text{ мс} - 7,05 \text{ мс}) \times 60 \text{ с/мин} \times 60 \text{ мин/ч} \times 24 \text{ ч/сут}}{1000 \text{ мс/с} \times 3600 \text{ с/ч}} = 0,0119 \text{ мА/ч}$$

Общее потребление в сутки:

$$\frac{3,088 \text{ мА/ч} + 0,0119 \text{ мА/ч}}{24 \text{ ч}} = 0,129 \text{ мА/ч}$$

Если питать устройство от двух последовательно соединенных 1,5 В батарей емкостью, например 3,1 А/ч, то время жизни батарей составляет 1000 дней.

Таким же образом можно рассчитать время жизни батарей и при других значениях интервала опросов.

Литература

1. Application Note AN053 Measuring power consumption with CC2430 & Z-Stack
2. Design Note DN106 Power Modes in CC111xFx, CC243x, and CC251xFx
3. Application Note AN044 CC2430 Revision D, Power Modes.

Ответственный за направление
в КОМПЭЛе – Мария Рудяк

Получение технической информации, заказ образцов,
поставка – e-mail: wireless.vesti@compel.ru



СИСТЕМА НА КРИСТАЛЛЕ **CC2430**

CC2430 – SoC, построенная на базе ZigBee-трансивера CC2420, объединенного с производительным контроллером 8051

Ключевые особенности:

- 8051 микроконтроллер с тактовой частотой 32 МГц
- Трансивер IEEE 802.15.4 (2.4 ГГц)
- 3 версии: 32/64/128 Кб программируемой FLASH-памяти
- Малое потребление энергии
- Статическая память 8 Кб (SRAM) с функцией DMA
- Один IEEE 802.15.4 MAC таймер, один обций 16-битный таймер и два 8-битных таймера
- Цифровая схема измерения уровня сигнала RSSI/LQI



Компэл
www.compel.ru