



*Microchip's Annual*  
**MASTERS**

# XLP

**Изучение возможностей управления  
энергопотреблением встраиваемых  
систем.**



# Микроконтроллеры с nanoWatt XLP

## Цели

### После данного класса, вы будете знать:

- | Разные режимы микропотребления, и управление ими.
- | Разницу технологий nanoWatt и nanoWatt XLP, а также режима Deep Sleep и других энергосберегающих режимов.
- | Анализ и проектирование системы с наименьшим энергопотреблением.



# Микроконтроллеры с nanoWatt XLP

## План

- | **Микропотребление**
- | **Планирование потребления.**
- | **Технология nanoWatt XLP**
- | **Deep Sleep**
- | **Итоги и Ссылки**



*Microchip's Annual*  
**MASTERS**

# Микропотребление

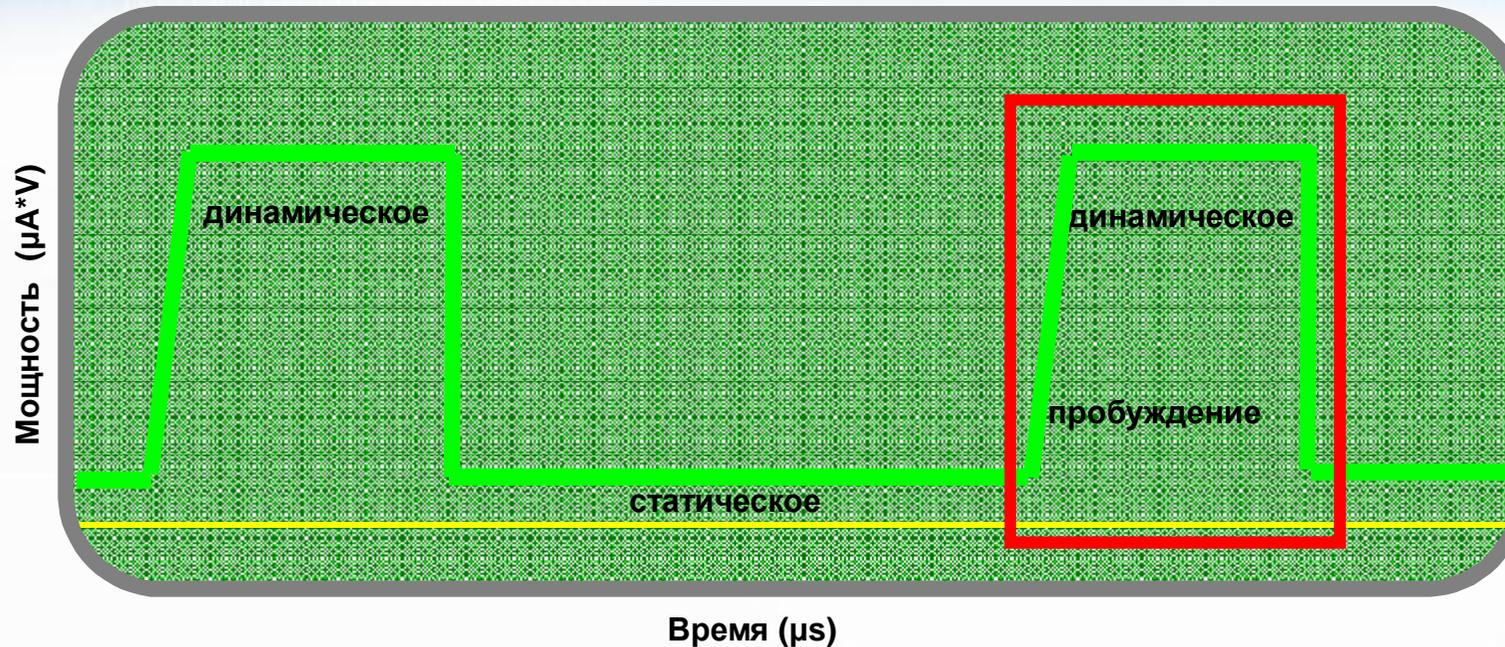


# Микропотребление Режимы Работы

- | **Существует несколько режимов работы микроконтроллера:**
  - | Динамический ( Активный)
    - | Системный генератор включен
    - | Паразитное, Генератор, Периферия, Ядро, Порты I/O
  - | Статический
    - | Системный генератор выключен
    - | Паразитное и Порты I/O
  - | Усреднённый
    - | Интегральное потребление за целый цикл приложения



# Микропотребление В Динамическом режиме работы



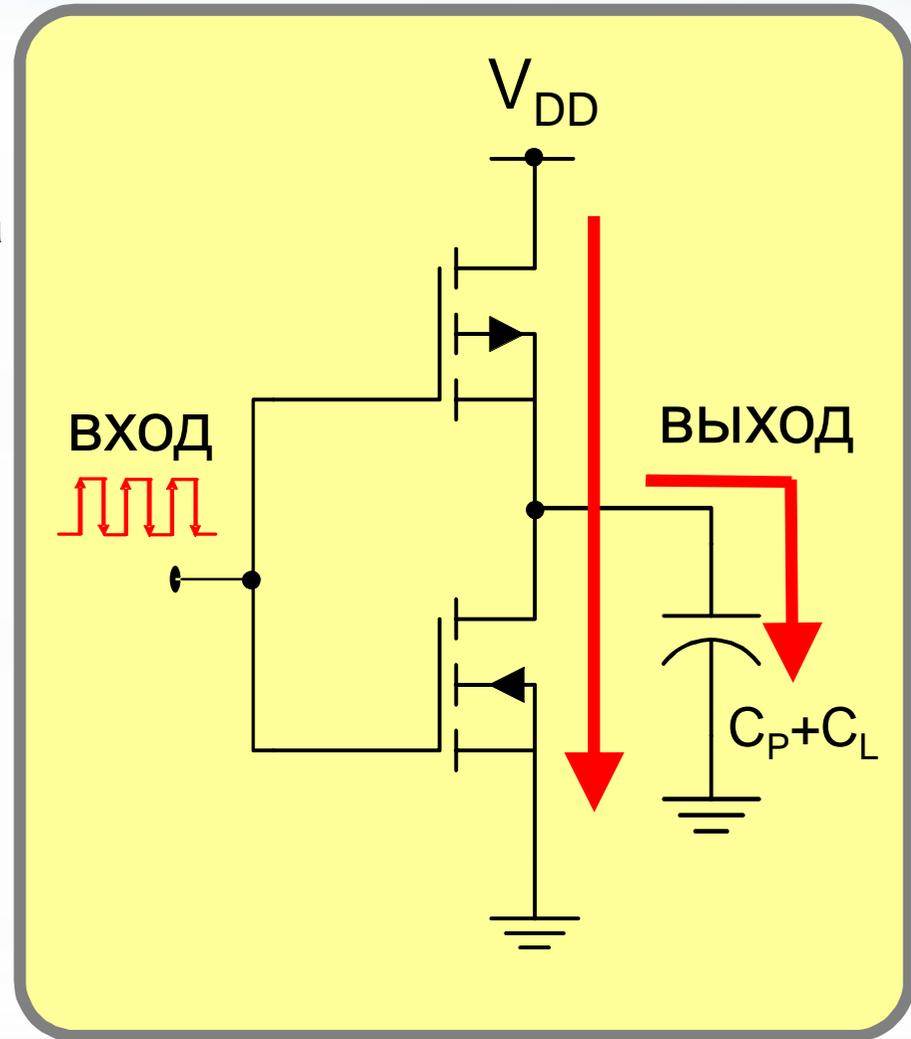
## Definition

**Динамическое (Активное) потребление** это мощность, потребляемая когда прибор активен и выполняет задачи. Эта мощность в основном уходит на переключение КМОП структур и является функцией частоты и напряжения. Дополнительная мощность потребляется периферией и портами I/O.



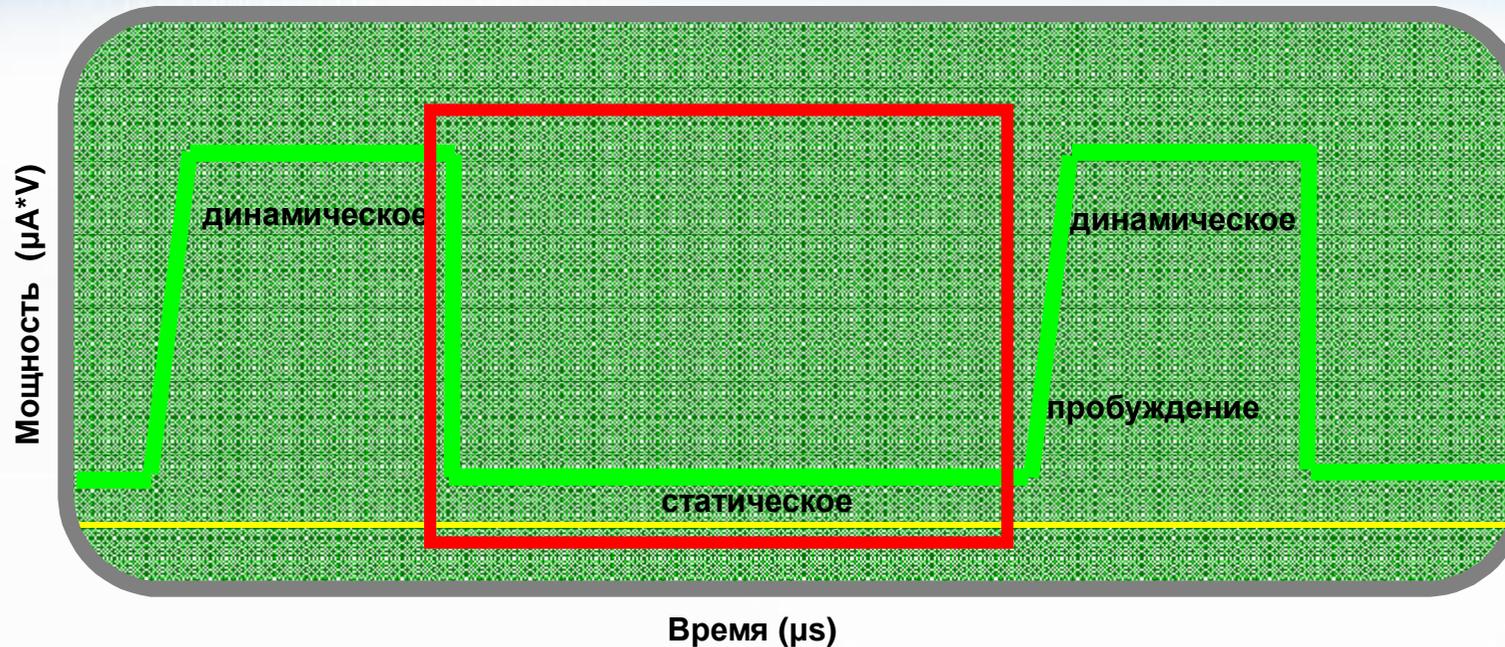
# Микропотребление Элементы динамического потребления

- ~ **Потери при переключении КМОП транзисторов**
  - ~ Во время переключения – оба открыты на некоторое время
  - ~ Частота переключения = больше вкл. = больше потерь
- ~ **Емкость затвора**
  - ~ Емкость нагрузки ( $C_L$ )
  - ~ Паразитическая емкость ( $C_P$ ) всегда около ~5-10pF
- ~ **Напряжение питания**
  - ~ Ниже питание – меньше потребление.





# Микропотребление В Статическом режиме работы

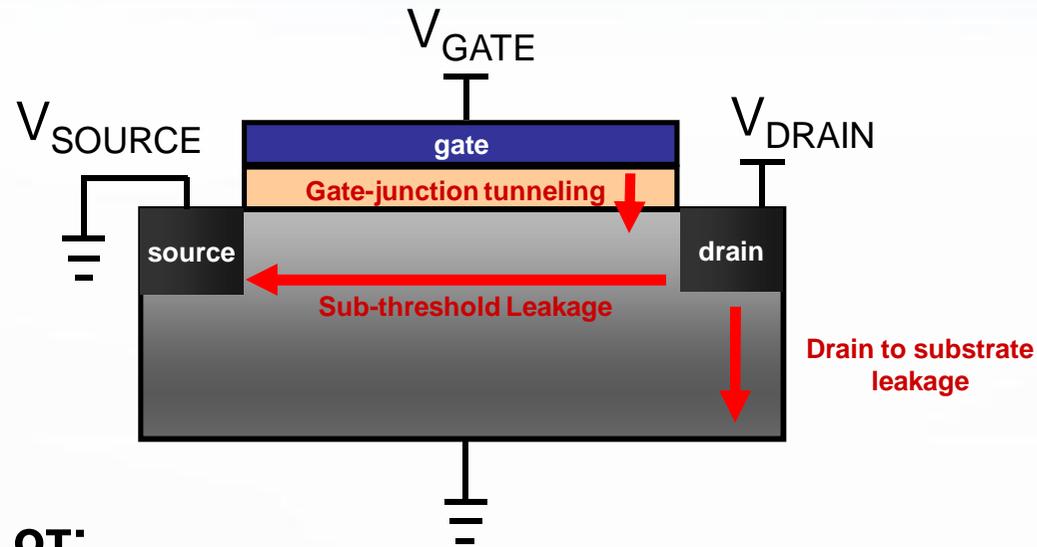


## Definition

**Статическое потребление** это потребление в режиме когда прибор включен, но не активен (например выключен генератор). Рассеивание энергии определяется утечкой в КМОП структурах, работой часов реального времени, супервизорами питания, сторожевым таймером, утечкой портов I/O и др.



# Микропотребление Элементы статического потребления



## Утечки зависят от:

- | Технологии процесса – меньше транзисторы, больше утечки
- | Напряжения – ниже напряжение, означает меньше утечки
- | Температуры – высокая температура увеличивает утечки



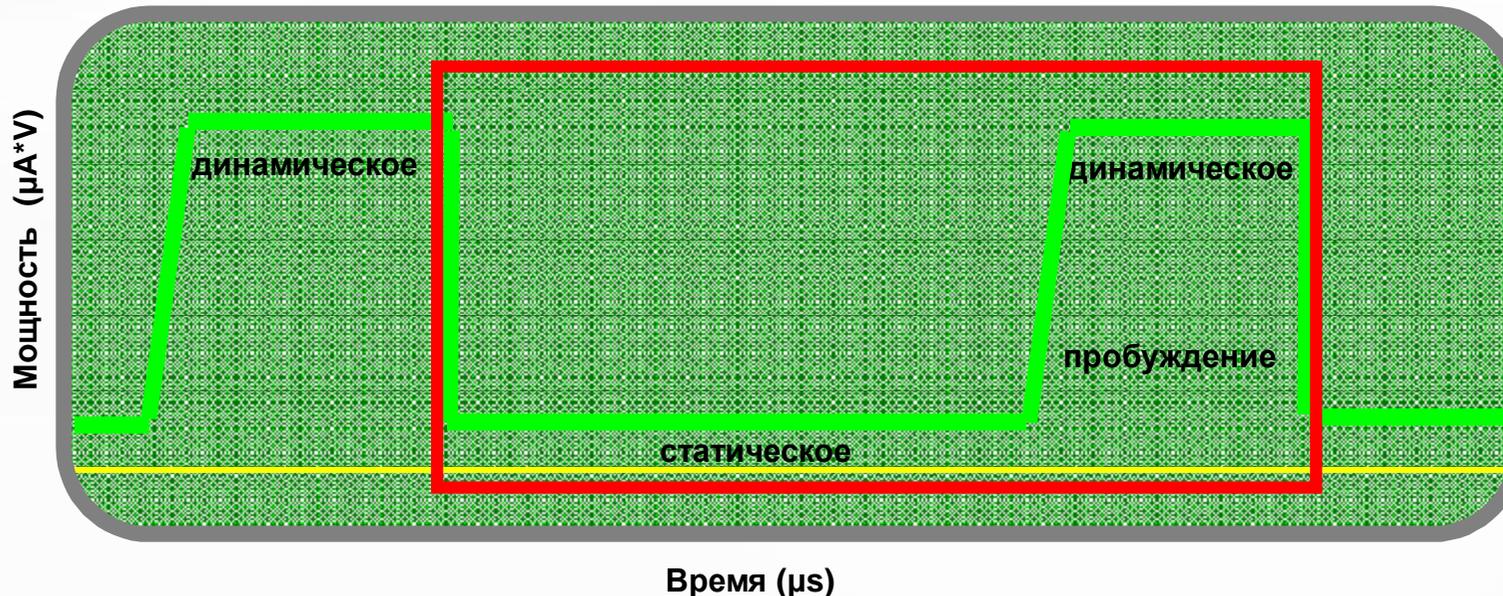
**Меньшие транзисторы увеличивают статическое потребление, но они могут работать на более низких напряжениях, тем самым уменьшая динамическое потребление**



# Микропотребление

## Усредненное потребление

$$P_{avg} = \frac{1}{t_{cycle}} \int (V_{active} \times I_{active} \times t_{active}) + (V_{static} \times I_{static} \times t_{static})$$



### Definition

**Усреднённая мощность** это мощность потреблённая за полный цикл работы устройства, работая как в активном, так и статическом режимах. Усредненная мощность включает в себя время проведённое в каждом из режимах, а также время переключения между активными и статическими режимами.



*Microchip's Annual*  
**MASTERS**

# Планирование потребления



# Планирование потребления Режимы работы.



## RUN [активное]

- | Ядро и Периферия тактируются на полной частоте
- | Ток **50-360µA/MHz** (3V, 25°C)
- | LP INTRC (31kHz) до **8µA** (1.8V, 25°C, PIC24F04KA201)



## DOZE (у некоторых контроллеров) [активное]

- | Ядро тактируется медленнее чем периферия, периферия работает на полной частоте.
- | Около **35%-75%** тока в режиме RUN



## IDLE (у некоторых контроллеров) [активное]

- | Ядро не тактируется, периферия ВКЛ.
- | Около **25%** от потребления в режиме RUN



## SLEEP [статическое]

- | Около **100nA** (3V, 25°C)
- | 85°C **1.35µA** (1.8V, 85°C, PIC24F04KA201)



## DEEP SLEEP (у некоторых контроллеров) [статическое]

- | SRAM, VREG, VBOR, RTCC – выкл.
- | Около **35nA** (3V, 25°C)

Увеличение срока  
работы от  
батарей



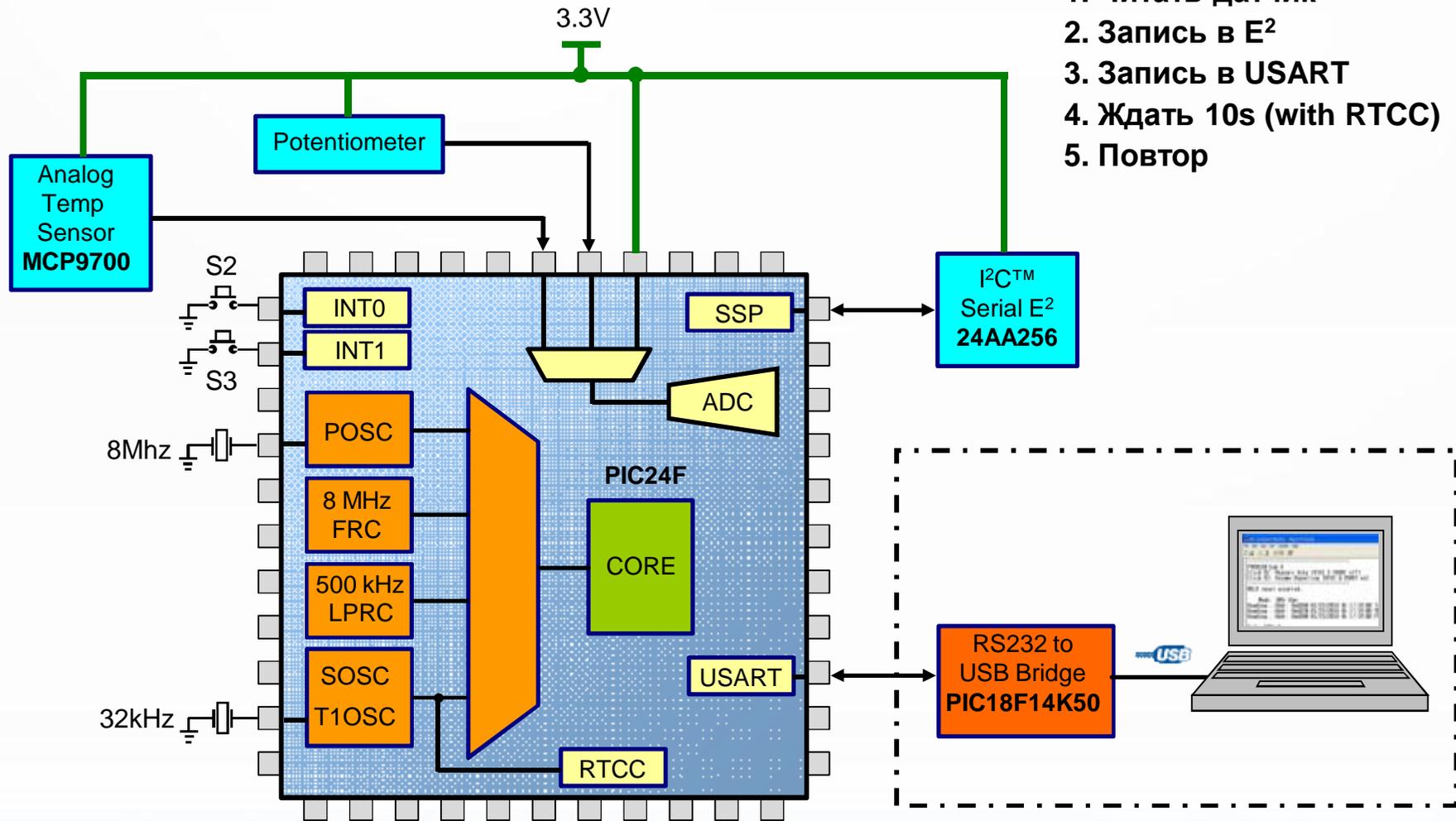
# Планирование потребления

## Анализ применения

- | **Разбить программу на фазы**
  - | Рассчитать ток потребляемый на каждой из фаз
  - | Вычислить время затрачиваемое на каждую из них
  - | Рассчитать общее потребление на фазах
  
- | **Рассчитываем среднюю мощность всего устройства**
  - | Может ли быть уменьшена, за счёт уменьшения времени в активном режиме?
  - | Можно ли изменить напряжение, тактовую частоту или режим микропотребления на какой-либо из фаз?
  
- | **Определяем худший вариант и пересмотреть**
  - | Возможно ли использование комбинаций разных режимов?



# Планирование потребления Блок схема

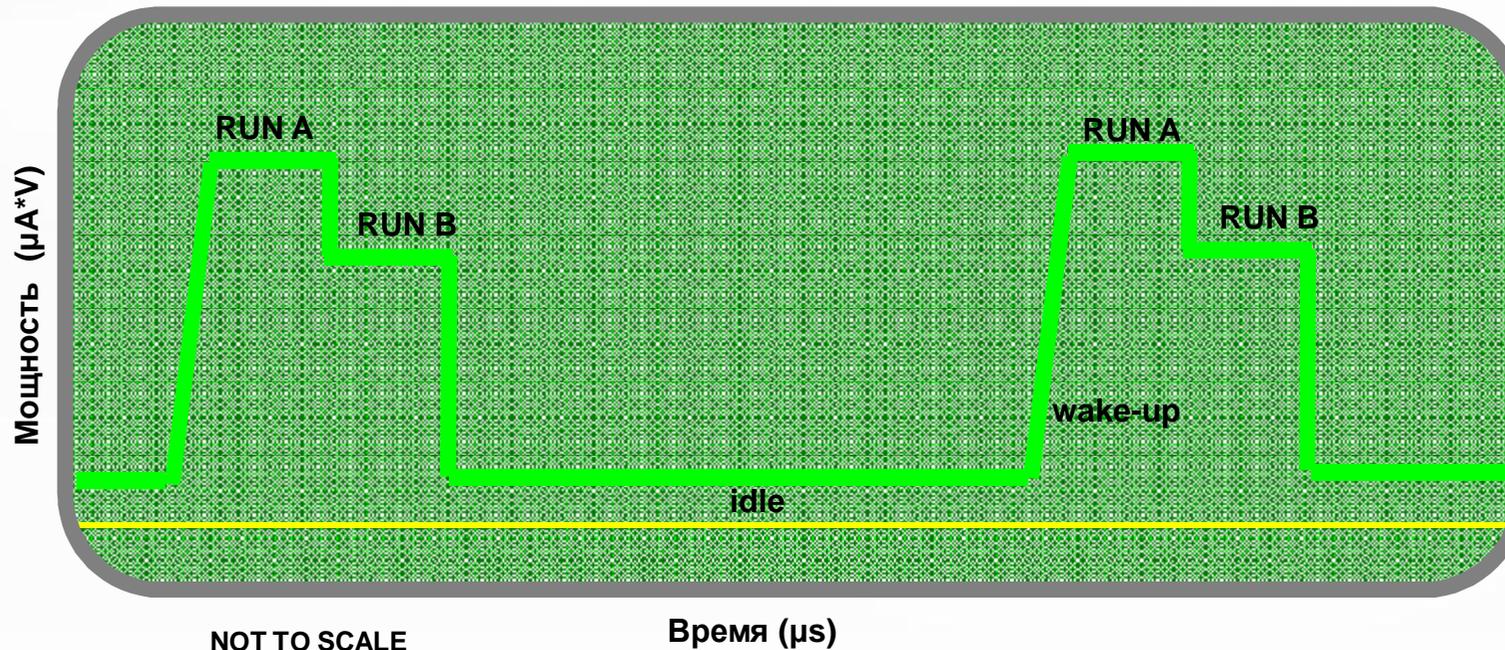


1. Читать датчик
2. Запись в E<sup>2</sup>
3. Запись в USART
4. Ждать 10s (with RTCC)
5. Повтор



# Планирование потребления Профиль потребления

*Минимизировать площадь под кривой*



- | Run A – измерить температуру ( $800\mu\text{s}$ )
- | Run B – запись в EEPROM ( $5\text{ms}$ )
- | Loop/Sleep/Idle/Deep Sleep на ( $10\text{s}$ )
- | Тактовая частота может меняться динамически
- | Рабочее напряжение может меняться динамически



# Планирование потребления *nanoWatt XLP Battery Life Estimator*



## **Выбрать МК и параметры**

Новые МК добавляются с новой версией, или вы сами можете их добавлять.  
На пример: "PIC18LFxxJ11.csv"

## **Выбрать батарейку**

Новые батарейки могут добавляться изменением файла  
"CustomBattery.csv"

## **Ввести режимы и потребление**

## **Пересмотреть и сохранить**

Microchip XLP Battery Life Estimator

Step 1: Select Device  
PIC24F16KA102 Device Notes

Step 2: Select Parameters  
Voltage: 3.3V Temperature: 25

Step 3: Select Battery

Description	Capacity	Self-discharge	Nominal V	Max cont. I	Max pulse I	Notes
LiMnO2 (CR2032)	225 mAh	0.08 %/mo(0.25 uA)	3.0 V	1.8 mA	12 mA	*
Li(CF)n (BR2032)	190 mAh	0.04 %/mo(0.106 uA)	3.0 V	0.2 mA	2 mA	*
LiFeS2 (AAA)	1150 mAh	0.05 %/mo(0.799 uA)	1.5 V	750 mA	750 mA	*
LiFeS2 (AA)	3000 mAh	0.05 %/mo(2.083 uA)	1.5 V	2000 mA	2000 mA	*
Li-ion (Approx. 'A')	2250 mAh	5 %/mo(0.156 mA)	3.6 V	2250 mA	2250 mA	*
Alkaline (AAA)	1150 mAh	0.3 %/mo(4.792 uA)	1.5 V	*	*	Drain rate affe
Alkaline (AA)	2850 mAh	0.3 %/mo(11.875 uA)	1.5 V	*	*	Drain rate affe

Step 4: Add/Modify Operational Mode

Label	Mode	Frequency	Time	Current	Charge	Peripherals
SLEEP w/RTCC	DEEP SLEEP	*	10 S/100%	0.835 uA	2.319 nAh	RTCC (w/ SOSC)

Add Modify Remove

Estimated Battery Life: 23 years 245 days 13 hours

Estimated Average Current: 0.835 uA

Estimated Peak Current: 0.835 uA

Load Settings Save Settings Create Report Quit



*Microchip's Annual*  
**MASTERS**

# Технология nanoWatt XLP



# nanoWatt XLP

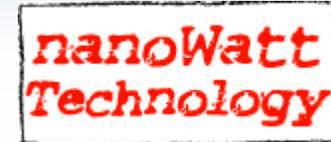
## Отличительные особенности

- | **Гибкость**
  - | Различные варианты тактирования
  - | Динамически изменяемая частота и источники
  - | Периферия с малым потреблением
  - | Конфигурация источников пробуждения с малым потреблением
- | **Малопотребляющие генераторы**
  - | Отсчет времени с WDT, Timer1 или RTCC без нарушения плана потребления
- | **Низкие утечки цифровых входов**
  - | Обычно < 50nA, иногда 5nA
- | **Малое время пробуждения**
  - | Уменьшается потеря времени на пробуждение



# nanoWatt XLP

## Технология nanoWatt (2003)



### Технология nanoWatt

- | Была представлена в 2003
- | Стандарт для всех МК Microchip с 2003
  
- | В покое (Sleep mode) мощность  $< 1\mu\text{W}$ 
  - | @3V  $I_{pd} < 333\text{nA}$  (PIC24H)
  - | @2V  $I_{pd} < 500\text{nA}$  (PIC16, PIC18, PIC24F)

### Definition

#### Технология nanoWatt

Набор методов проектирования и разработки микроконтроллеров, обеспечивающий возможность потребления микроконтроллером менее  $1\mu\text{W}$  в ждущем режиме ( $I_{pd}$ ).



# nanoWatt XLP

## Технология nanoWatt (2003)



- | **Режим IDLE**
  - | CPU Выкл, Периферия Вкл
- | **Встроенный, высокоскоростной RC генератор (INTRC) с PLL и программируемым пост- делителем позволяет нам:**
  - | Быстрый старт за  $1\mu\text{s}$ - $5\mu\text{s}$
  - | Двухскоростной старт (старт на INTRC, затем переход на кварц, если нужно)
  - | Динамически переключаемая частота
- | **Расширенный интервал WDT**
  - | Максимальные тайм-аут увеличен с  $18\text{ms}$  до  $131\text{s}$
- | **Timer1 с низким потреблением (TMR1) и 32 kHz второй генератор (SOSC)**
- | **Малопотребляющий программно контролируемый BOR**
  - | Стандартный BOR был переработан для меньшего потребления
  - | Программно можно выключать его, когда не нужен (например в Sleep)



# nanoWatt XLP

## Технология *nanoWatt XLP* (2009)

### | nanoWatt XLP (eXtreme Low Power)

- | Представлена в 2009 как новое поколение nanoWatt
- | Транзисторы с меньшими утечками тока
- | Стандарт для новых МК Microchip
- | Специальная малопотребляющая периферия

	Requirements	Best specifications achieved to date
Sleep:	100nA или <u>менее</u>	13nA @1.8V
Real-Time Clock Calendar (RTCC):	800nA или <u>менее</u>	500nA @1.8V
Watchdog Timer (WDT):	800nA или <u>менее</u>	200nA @2.0V

### Definition

#### Технология nanoWatt XLP (eXtreme Low Power)

Технология Microchip используемая для дизайна микроконтроллеров с потреблением тока менее 100nA в режиме ожидания, 800nA с запущенным RTCC и 800nA с запущенным WDT.



# nanoWatt XLP

## Технология *nanoWatt XLP* (2009)

### Обзор специализированных особенностей управления питанием:

- | Deep Sleep (DS)
- | Deep Sleep Brown-Out Reset (DSBOR)
- | Deep Sleep Watchdog Timer (DSWDT)
- | Низкий ток утечки входов
- | Спецификации при 60°C для работы с батарейками.



#### ~ **Пример питания от батареи (25°C)**

~ *Battery Type: Coin Cell (CR2032)*

~ *1мс RUN при 1MHz, затем Deep Sleep с включенным RTCC*

~ **PIC24FXXKA с *nanoWatt XLP* - До 20 лет**



# nanoWatt XLP

## Контроллеры с двумя *Brown-Out Resets*

### | **Brown-Out Reset (BOR)**

- | Стандартный nanoWatt BOR
- | В некоторых МК назван LPBOR
- | Конфигурируется на 4 уровня напряжения
- | Обычное потребление тока  $\sim 5\mu A$



### | **Deep Sleep BOR (DSBOR)**



- | Доступен в дополнение к BOR в контроллерах с режимом Deep Sleep
- | В некоторых МК до 5nA



# nanoWatt XLP

## Контроллеры с двумя Watchdog Timers

- | Watchdog Timer (WDT)
  - | Стандартный nanoWatt WDT
  - |  $I_{\Delta WDT}$  до 500nA
  - | Период тайм-аута: 1ms–131s



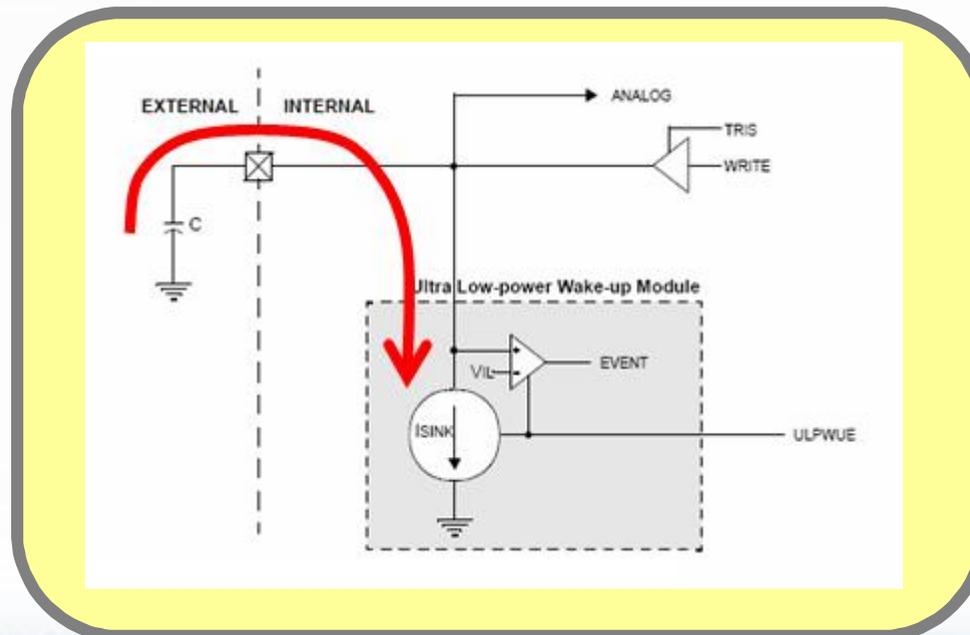
- | Deep Sleep WDT (DSWDT) 
  - | Доступен в дополнение к WDT на МК с режимом Deep Sleep
  - | Применим для приложений, которые находятся долгое время неактивные
  - | DSWDT идет в режиме Deep Sleep при
  - |  $I_{\Delta WDT}$  до 370nA
  - | Период тайм-аута: 2.1ms-25.1 days



# nanoWatt XLP

## модуль *Ultra Low-Power Wake-Up (ULPWU)*

- | Стандартный модуль на большинстве МК
- | 75nA-160nA потребляемый ток
- | Сравнимый по тайм-ауту с DSWDT, но на 80% меньше потребления тока
- | Не плавает и не потребляет так много тока, как стандартный I/O порт
- | Время пробуждения зависит от температуры и влажности





# nanoWatt XLP Peripheral Module Control

- | **Биты разрешения периферии**
  - | Находится в регистре контроля периферией SFRs
    - | Разрешает/запрещает функционирование этой периферии
    - | Управляющие регистры доступны для чтения и записи
    - | Пример: AD1CON1<ADON>
- | **Некоторые контроллеры имеют в добавок дополнительные биты отключения периферии Peripheral Module Disable (PMD)**
  - | Находятся в PMD регистрах
    - | Отключает тактирование периферии
    - | Убирает питание от регистров периферии
    - | Пример: PMD1<ADC1MD>

TABLE 4-23: PMD REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets
PMD1	0770	—	—	T3MD	T2MD	T1MD	—	—	—	I2C1MD	U2MD	U1MD	—	SPI1MD	—	—	ADC1MD	0000
PMD2	0772	—	—	—	—	—	—	—	IC1MD	—	—	—	—	—	—	—	OC1MD	0000
PMD3	0774	—	—	—	—	—	CMPMD	RTCCMD	—	CRCPMD	—	—	—	—	—	—	—	0000
PMD4	0776	—	—	—	—	—	—	—	—	—	—	—	EEMD	REFOMD	CTMUMD	HLVDMD	—	0000



# nanoWatt XLP

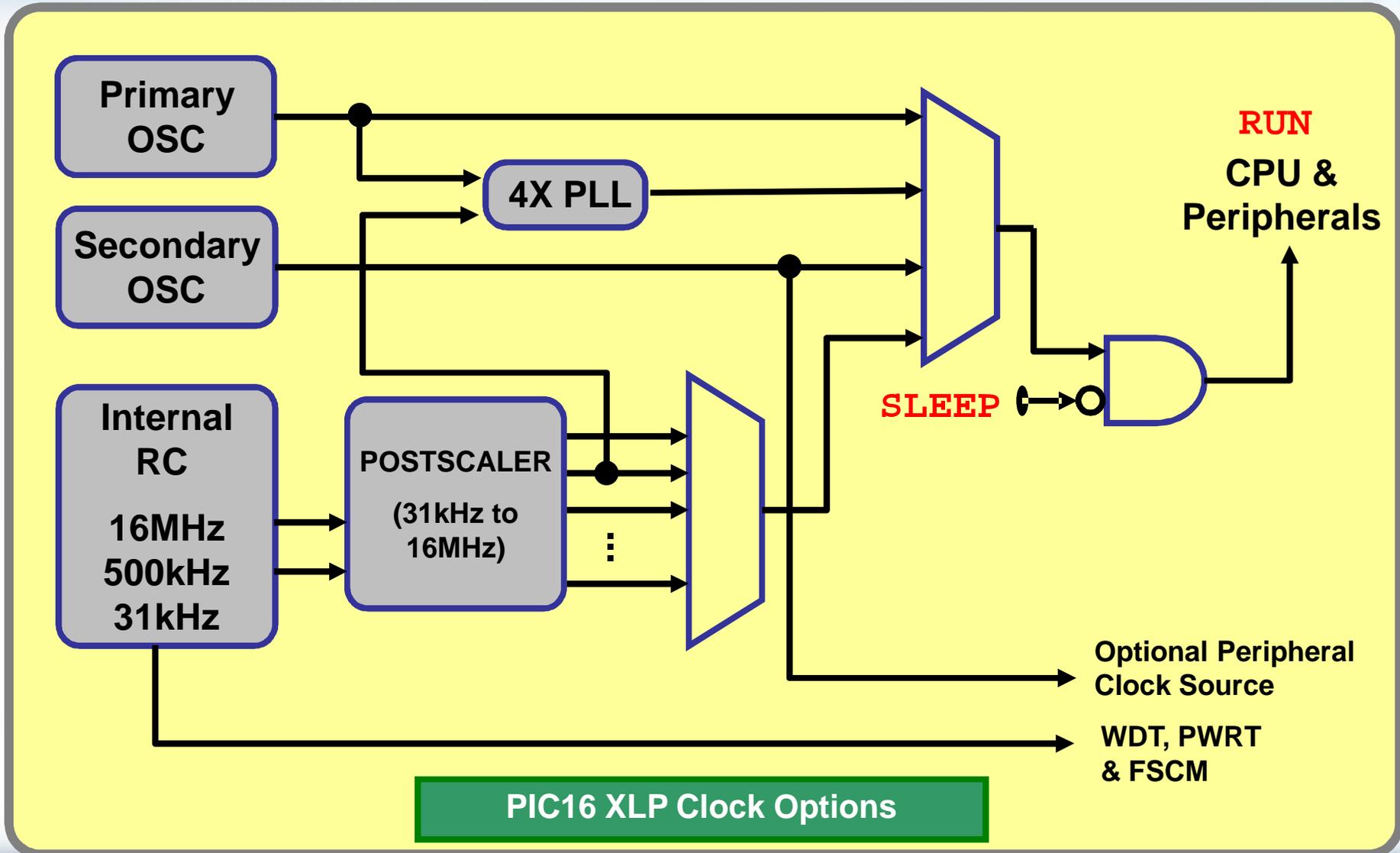
## Рабочие режимы

	PIC16	PIC18	PIC24
<b>RUN</b> <i>Все работает в активном режиме</i>	ü	ü	ü
<b>DOZE</b> <i>CPU медленнее чем периферия</i>			ü
<b>IDLE</b> <i>CPU выкл, периферия вкл</i>		ü	ü
<b>SLEEP</b> <i>Системный генератор выкл</i>	ü	ü	ü
<b>DEEP SLEEP</b> <i>RAM выкл, Vreg выкл</i>		-	-

- Только в МК nanoWatt XLP

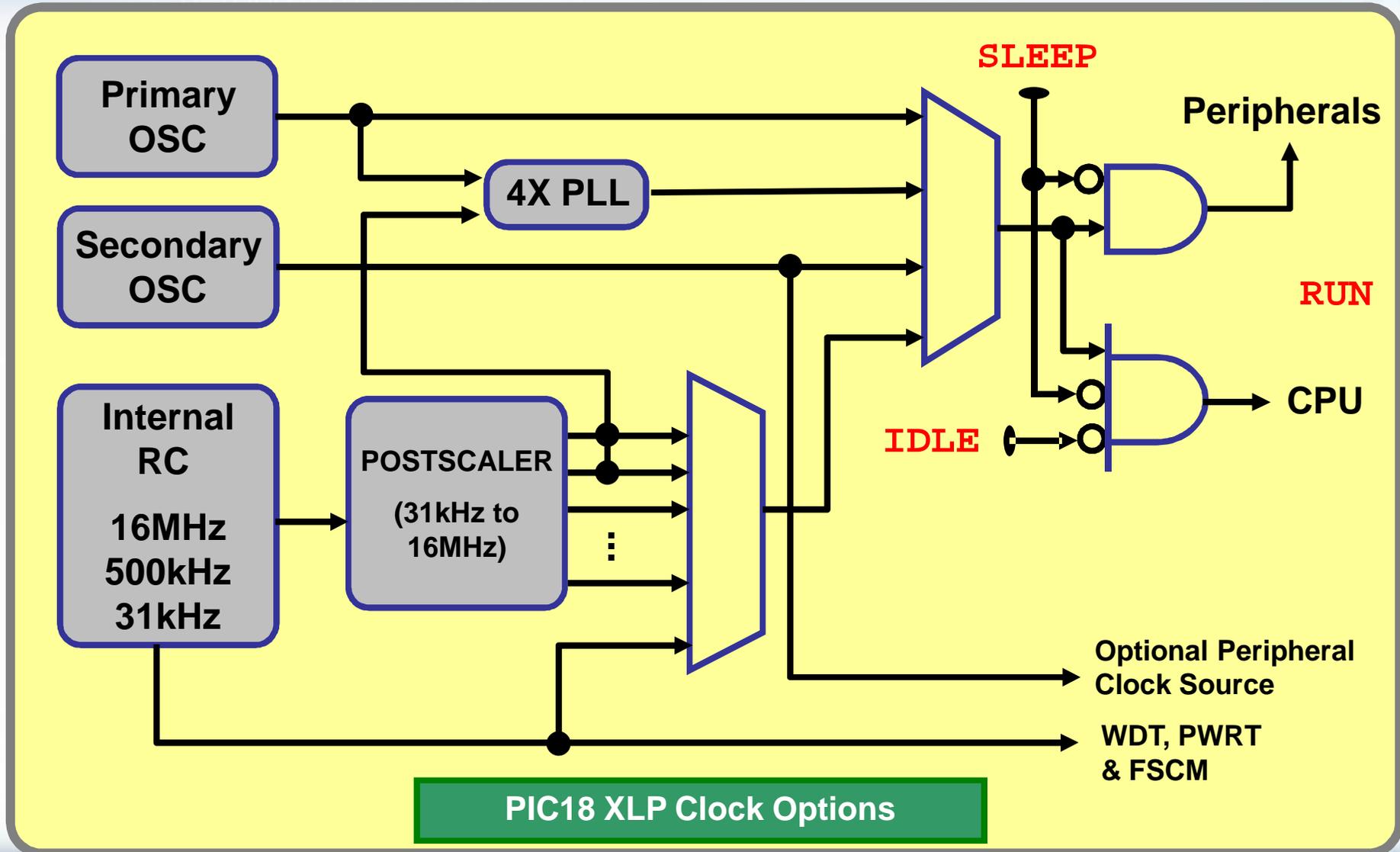


# nanoWatt XLP PIC16 XLP Clock Options





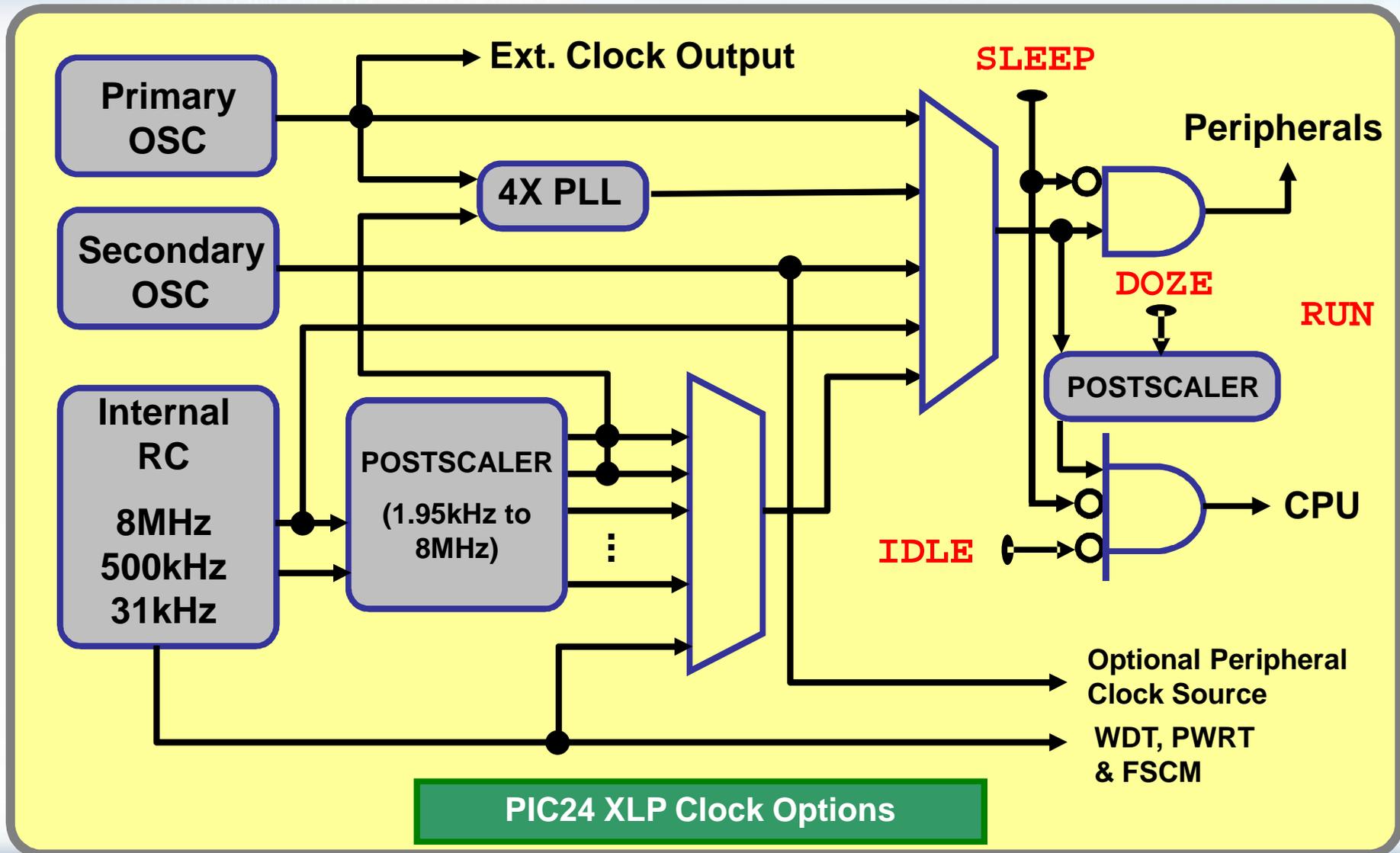
# nanoWatt XLP PIC18 XLP Clock Options



PIC18 XLP Clock Options

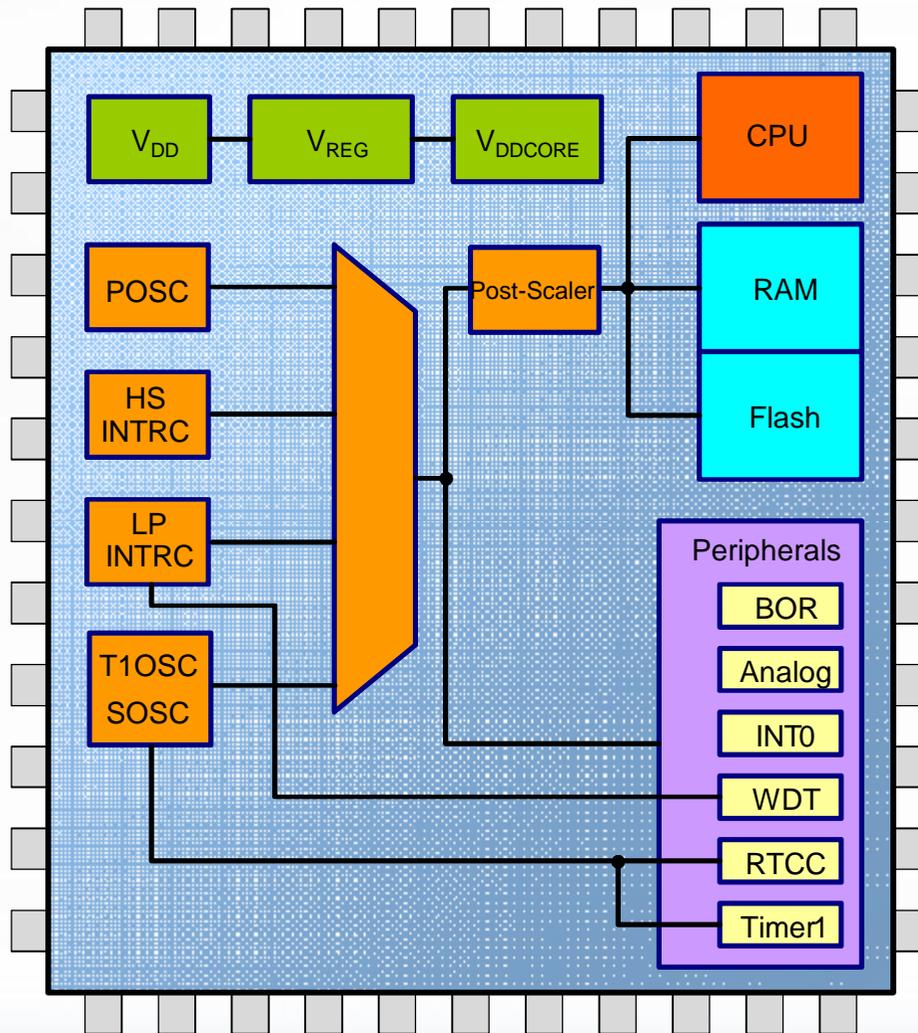


# nanoWatt XLP PIC24 XLP Генератор





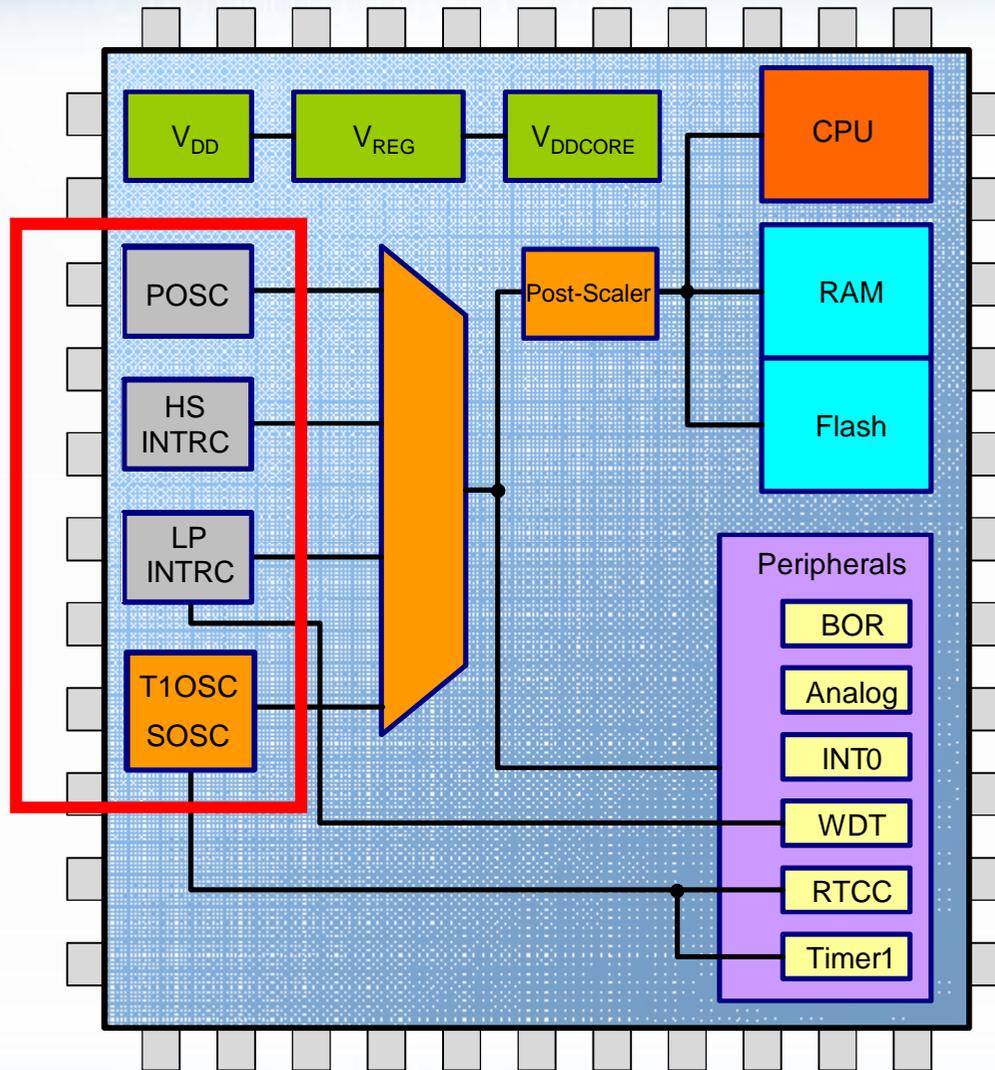
# nanoWatt XLP Run Mode



- ~ Все ресурсы активны
- ~ Динамически конфигурируемый системный генератор



# nanoWatt XLP Clock Switching



- ~ Несколько источников тактирования
- ~ Может переключаться динамически
- ~ Влияет на все части МК



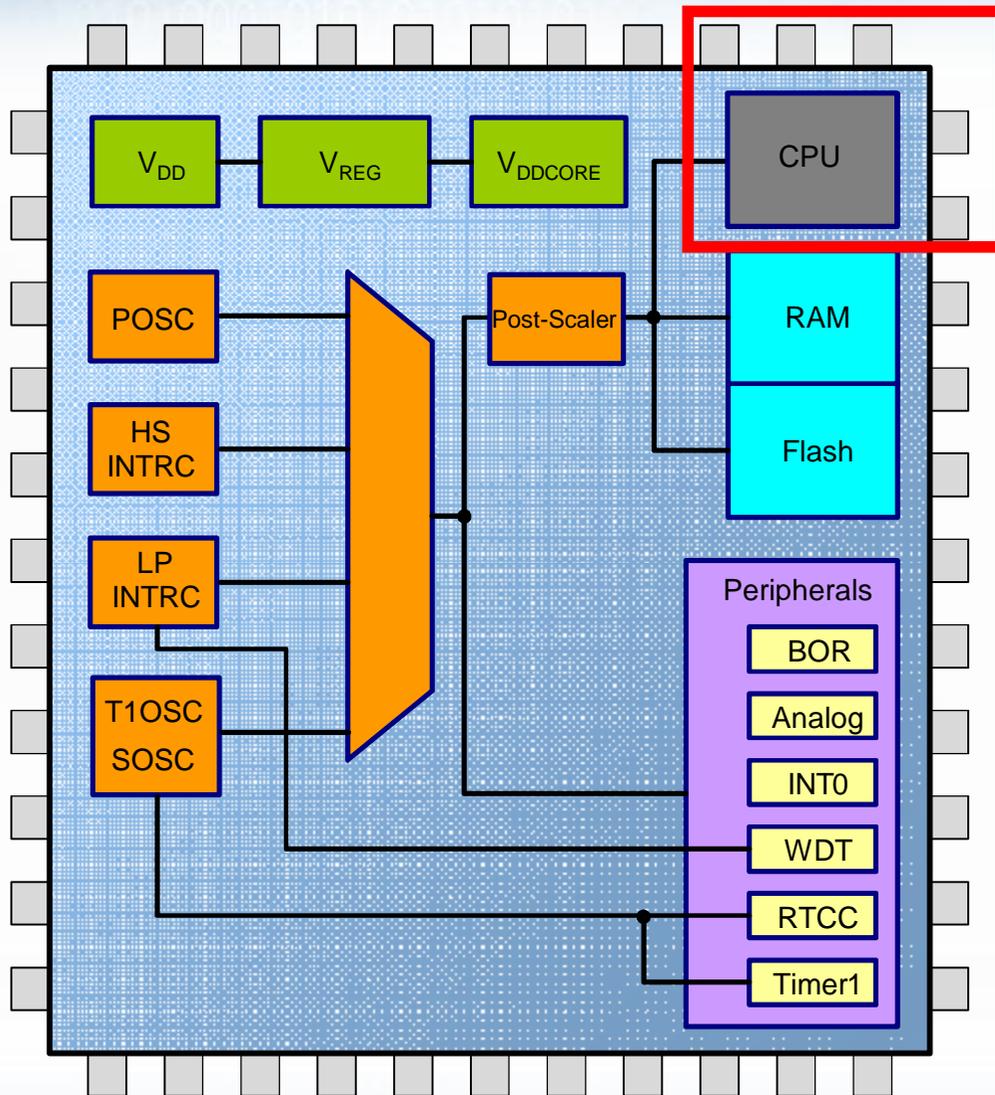
# nanoWatt XLP Clock Switching

- | **Смена частоты может сохранить больше энергии чем режимы Idle и Doze**
  - | Пониженная частота влияет на все части МК
- | **Двойной запуск**
  - | Старт на INTRC за 1 $\mu$ s-5 $\mu$ s
  - | Переключаемся на кварц, если надо
  - | Работаем на INTRC пока ждем PLL
- | **Уменьшение частоты используется пока ждем внешних событий или медленную периферию такую как АЦП, компаратор, коммуникационный порт, и т.д.**





# nanoWatt XLP Idle Mode



- ~ CPU  
ВЫКЛЮЧЕН
- ~ Периферия  
включена
- ~ 25%  
потребления  
от Run



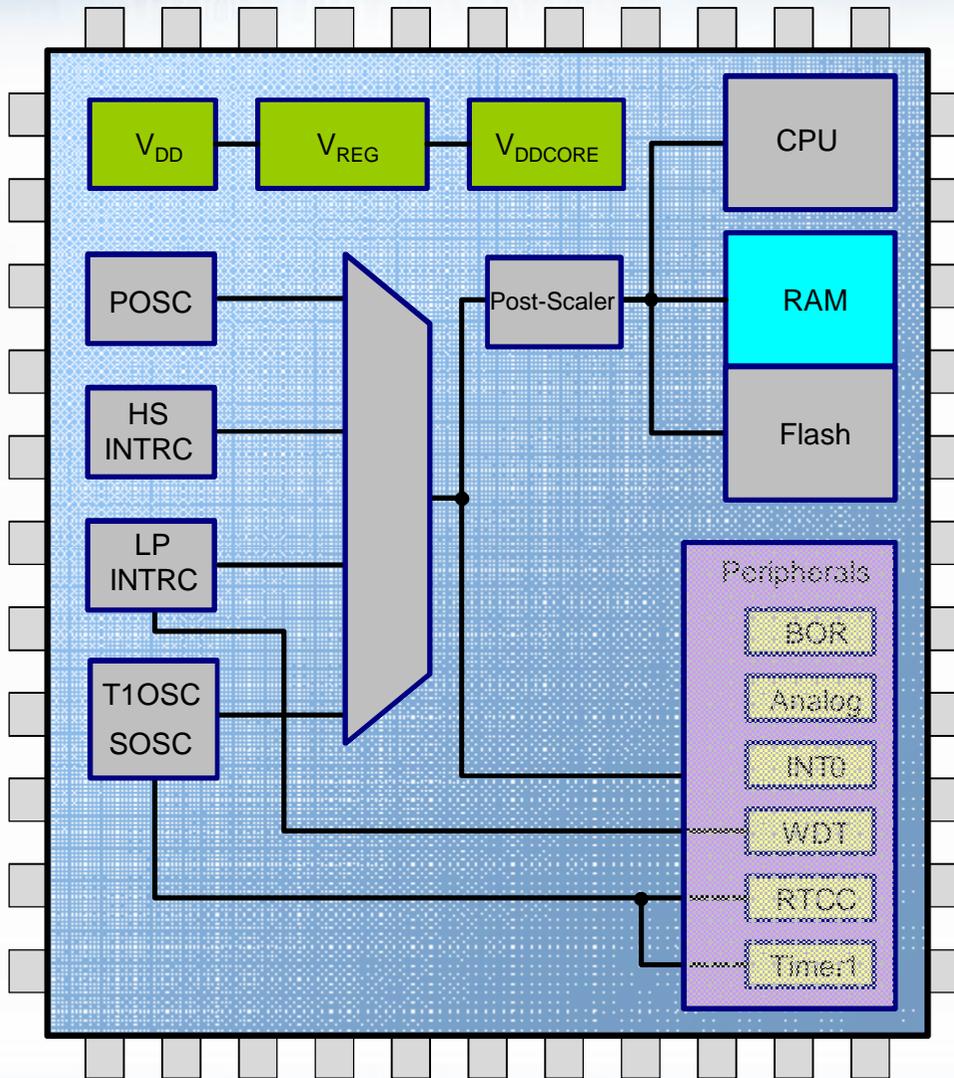
# nanoWatt XLP

## *Idle u Doze Modes*

- | **Когда могут режимы Idle или Doze режимы использоваться?**
  - | **Заменять циклы `while(!Interrupt)`**
  - | **Замедляемся пока ждём периферию или прерывание**
  - | **Когда нужна быстрая реакция на событие и быстрое пробуждение**
    - | **Пробуждение обычно за 1 инструкцию**
  - | **Во время передач по DMA**
  - | **Когда приложение постоянно опрашивает что-либо или находится на связи**



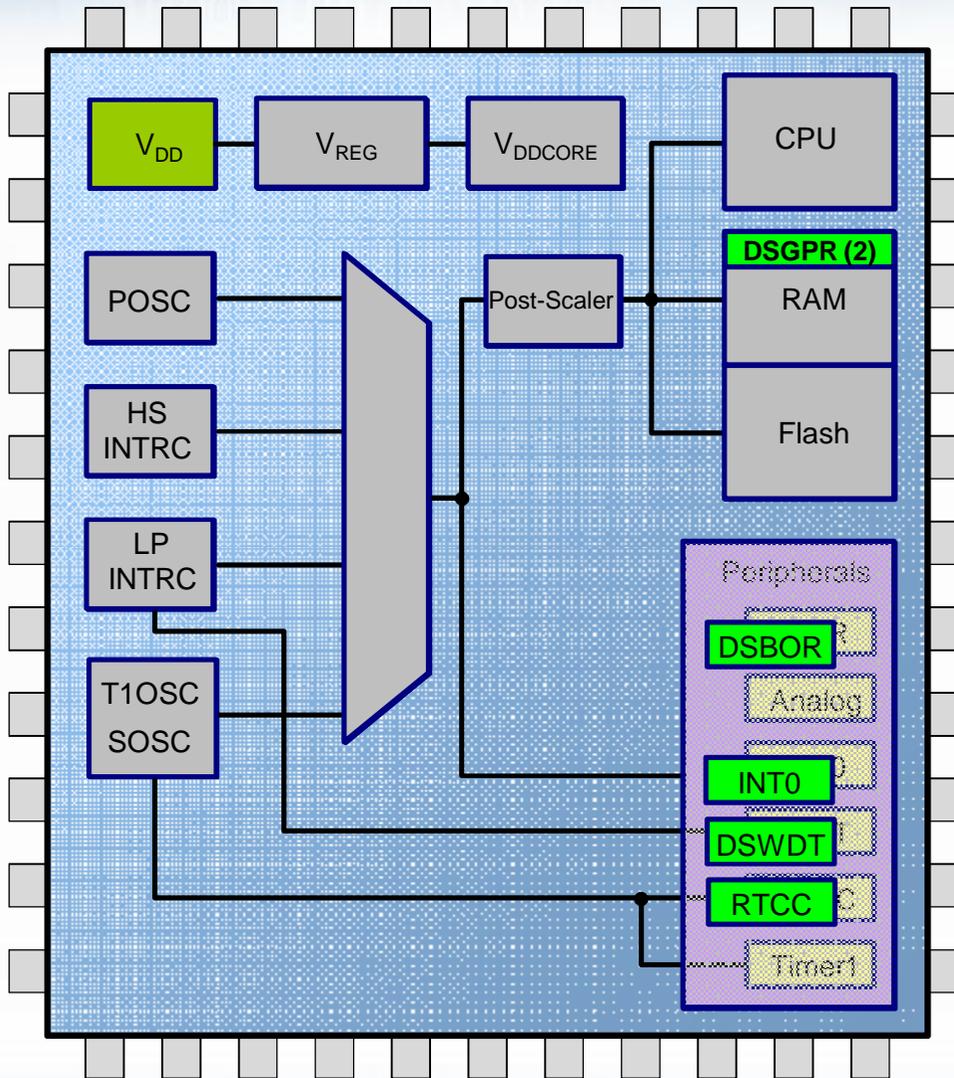
# nanoWatt XLP Sleep Mode



- ~ **50-100nA** без регулятора
- ~ **3-5 $\mu$ A** с внутренним регулятором
- ~ Генератор, CPU выключены
- ~ RAM остается запитанной
- ~ Регуляторы остаются включенными
- ~ Некоторая периферия может работать в режиме Sleep



# nanoWatt XLP Deep Sleep Mode



- ~ **<50nA**
- ~ **RAM не питается**
- ~ **Регуляторы выключены**
- ~ **Некоторая периферия продолжает работать в Deep Sleep**
  - ~ DSBOR
  - ~ DSWDT
  - ~ RTCC
  - ~ INTO



# nanoWatt XLP *Deep Sleep Mode*

- | **Питание ядра отключено**
  - | Потеря данных RAM, SFRs и Программного счётчика
  - | Два DSGPR регистра хранят данные в Deep Sleep
  - | Время пробуждения включает время на стабилизацию напряжения внутреннего регулятора
- | **Пробуждение вызывает Power-On Reset**
  - | Стандартный Sleep
    - | Продолжает выполнение программы после Sleep
  - | Deep Sleep
    - | Пробуждение с чистым программным счетчиком
- | **Порты I/O не меняют состояние**



# nanoWatt XLP

## Итог таблица из AN1267

Modes	Active Clocks	Active Peripherals	Wakeup Sources	Typical Current	Typical Usage
<b>RUN</b>	All	All			
<b>DOZE</b>	All	All	All Software wake-up	~50% of Run Current	Applications with high-speed peripherals requiring low CPU use
<b>IDLE</b>	Peripheral Clocks Timer1 Secondary OSC INTRC LPRC ADC RC	All	All	~25% of Run Current	Anytime device is waiting for an event
<b>SLEEP</b>	Timer1 Secondary OSC INTRC LPRC ADC RC	RTCC Timer1 WDT INTx BOR ADC HLVD CVREF Comparators UART-RX	All	50-100 nA base  3-5 uA with Internal Regulator	Most low-power apps
<b>DEEP SLEEP</b>	Secondary OSC LPRC	RTCC DSWDT DSBOR INT0	RTCC DSWDT DSBOR INT0 MCLR ULPWU	< 50 nA base  Peripherals add incremental current	Long-life battery based applications, applications with long sleep times



# nanoWatt XLP

## Сравнение потребления

	PIC16LF1827	PIC16LF1937	PIC16LF727	PIC18LF14K22	PIC18LF14K50	PIC18LF46J11	PIC18LF46J50	PIC18LF46K20	PIC24F04KA201	PIC24F16KA102	PIC24FJ64GA104	PIC24FJ64GB004
Deep Sleep (nA)						13	15		20	20	20	20
Sleep (nA)	20	60	20	34	24	54	60	100	25	25	200	200
WDT (nA)	500	500	500	460	450	820	780	600	400	400	200	200
32kHz SOSC/RTCC (nA)	600	600	600	650	790	850	830	600	500	500	500	500
1 MHz Run (µA)	80	93	80	131	125	275	275	131	195	195	250	250
Minimum V <sub>dd</sub> (V)	1.8	1.8	1.8	1.8	1.8	2.0	2.0	1.8	1.8	1.8	2.0	2.0



Current specifications are typical (TYP) values at minimum V<sub>dd</sub>



*Microchip's Annual*  
**MASTERS**

# Deep Sleep





# Deep Sleep

## Что такое Deep Sleep (DS)?

- | **Выключает питание у:**
  - | Ядра, периферии, SRAM и регулятора напряжения
- | **Наименьшее потребление:**
  - | В лучшем случае **13 nA**
  - | RTCC в течении DS до **500 nA**
- | **Порты I/O остаются запитаны и сохраняют свои состояния в Deep Sleep**
- | **Некоторая периферия продолжает работать и может пробуждать из Deep Sleep**

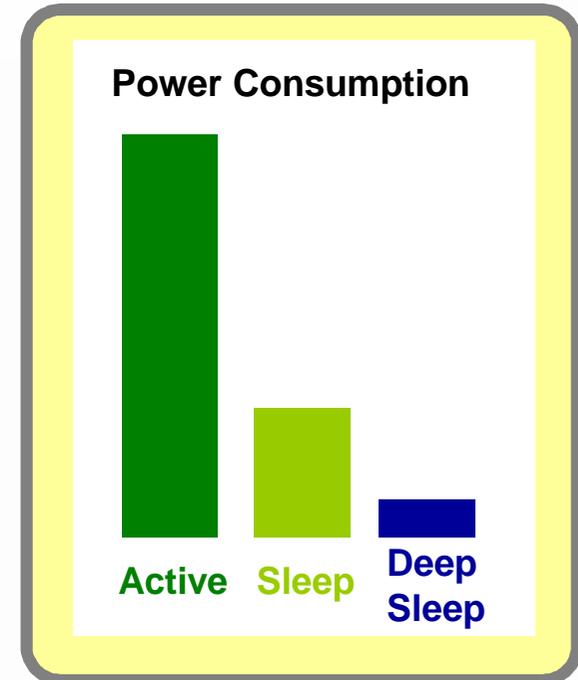




# Deep Sleep

## Отличия от Sleep?

- | Потребление на 90% меньше чем Sleep
- | SRAM выключена
  - | Регистры HE сохраняются в DS
- | Специальные регистры для сохранения данных
  - | Два регистра сохраняют данные в DS
  - | FLASH или EEPROM можно также использовать для хранения данных
- | Внутренний регулятор напряжения (LDO) выключается
  - | Не у всех МК есть LDO
- | Пробуждение из DS вызывает Power-On Reset
  - | Выполнение программы продолжается с вектора RESET
  - | SFRs сбрасываются на стандартные значения
  - | RCON<DPSLP> устанавливаются аппаратно

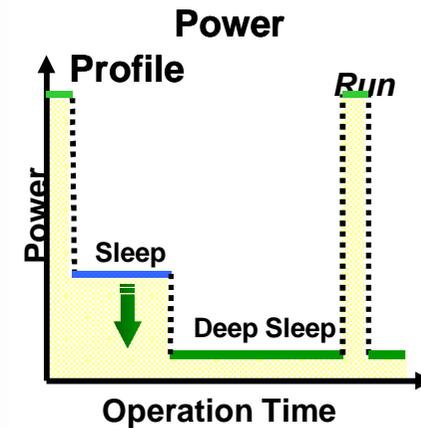




# Deep Sleep

## Таблица сравнения Sleep и Deep Sleep

Low Power Mode	SLEEP	DEEP SLEEP
<b>Определение</b>	Ядро не питается, Некоторая периферия работает, RAM сохр.	Ядро, периферия, SRAM и регуляторы напряжения не питаются.
<b>Ресурсы пробуждения</b>	RTCC Watch-Dog Timer Brown-out Reset Interrupt Pins ULPWU Power-On Reset Reset Pin (MCLR) Peripherals UART (RX)	DS RTCC DS Watch-Dog Timer DS Brown-out Reset INT0 ULPWU Power-On Reset Reset Pin (MCLR)
<b>Пробуждение</b>	Около (~1μ-5μS typ)	Дольше (same as POR)
<b>Pin State</b>	Сохранены	Сохранены
<b>RAM State</b>	Сохранены	2 регистра сохранены





# Deep Sleep

## Когда DS эффективен?

### Когда приложение:

- Неактивно большую часть времени
  - Обычно больше чем 1с
- Требует точные временные интервалы с минимальным потреблением
- Выполняется на экстремальных температурах
- Не требует много периферии (только периферию способную работать в DS)

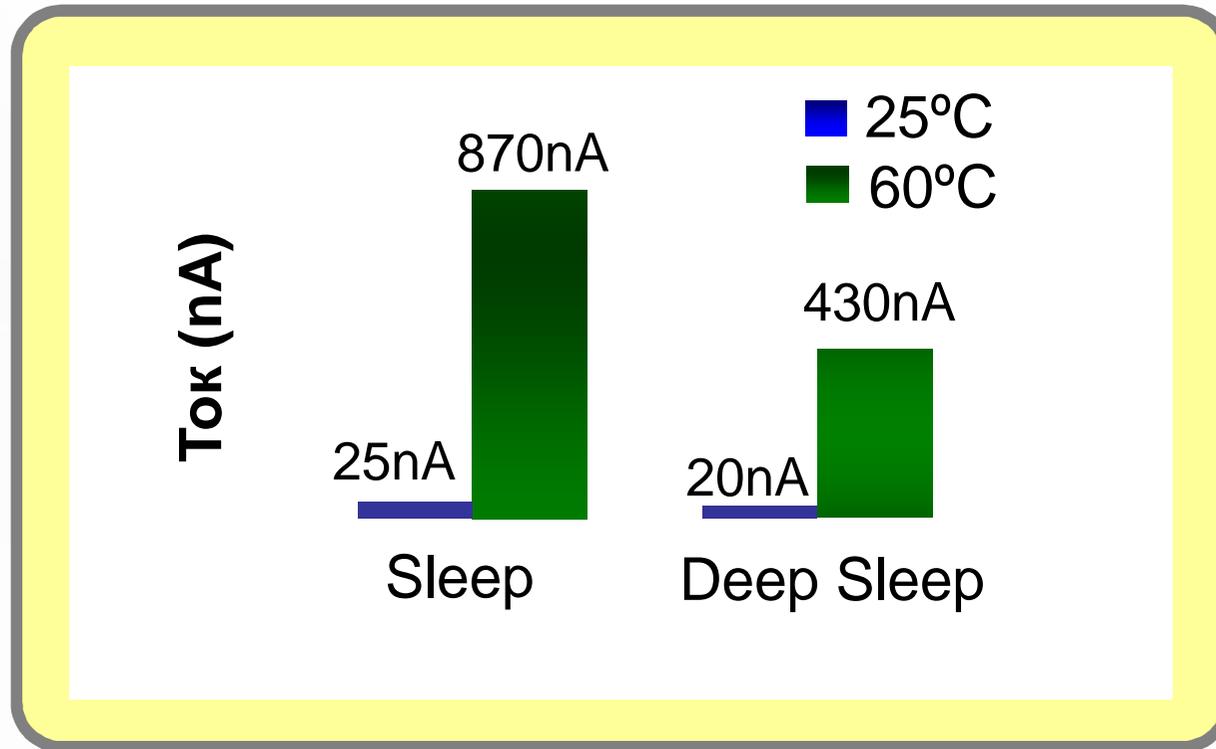




# Deep Sleep

## Когда DS эффективен?

В приложениях с высокой температурой:



### PIC24F16KA102

25°C specifications - Typ  $I_{pd}$  @ 1.8V

60°C specifications - Max  $I_{pd}$  @ 1.8V



# Deep Sleep

## Что работает в DS?

### RTCC

- | Продолжает считать время
- | RTCC может выдавать импульс каждую секунду



### I/O Pins

- | Сохраняют состояние

### Специальные регистры Deep Sleep сохраняют значения:

- | DSGPR0 – DS Регистр общего назначения 0
- | DSGPR1 – DS Регистр общего назначения 1
- | RTCC – Real-Time Clock Calendar

### DSBOR (Deep Sleep Brown-Out Reset)

- | Мониторинг  $V_{DD}$  в течении DS

### DSWDT (Deep Sleep Watchdog Timer)



# Deep Sleep *Break-Even Time*



## Break-Even Time

- | DS отключает ядро, SRAM и регулятор напряжения
- | При просыпании после DS:
  - | Включается регулятор напряжения
  - | Выполняется процедура POR
  - | Запускается генератор
  - | Восстанавливается контекст
- | Время пробуждения становится доминантным
- | Для коротких пауз, стандартный режим sleep может быть лучшим выбором

### Definition

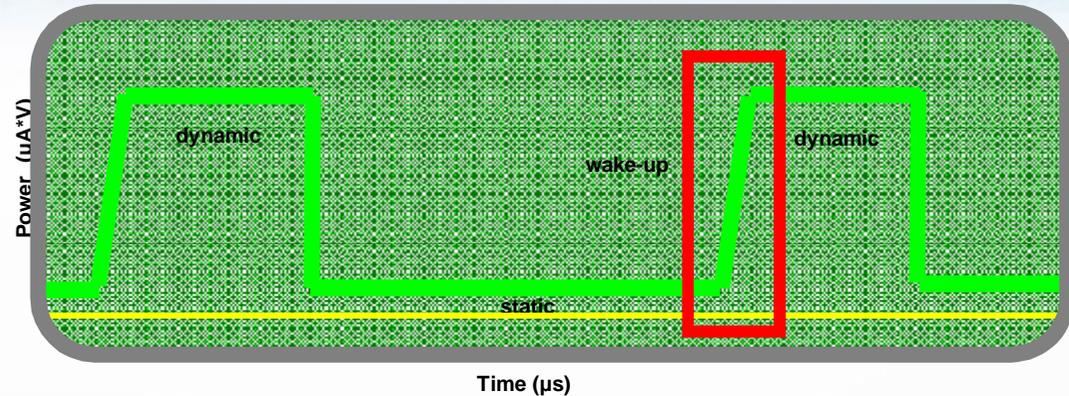
#### **Break-Even Time**

Точка, когда Deep Sleep потребляет меньше чем Sleep



# Deep Sleep

## Составляющие времени пробуждения



### Запуск регулятора

- | Около  $10\mu s$
- | Требуется для МК со встроенным LDO

### Power-up Time

- | Около  $72ms$

### Запуск генератора

- | Резонатор может стартовать  $100\mu s - 200\mu s$

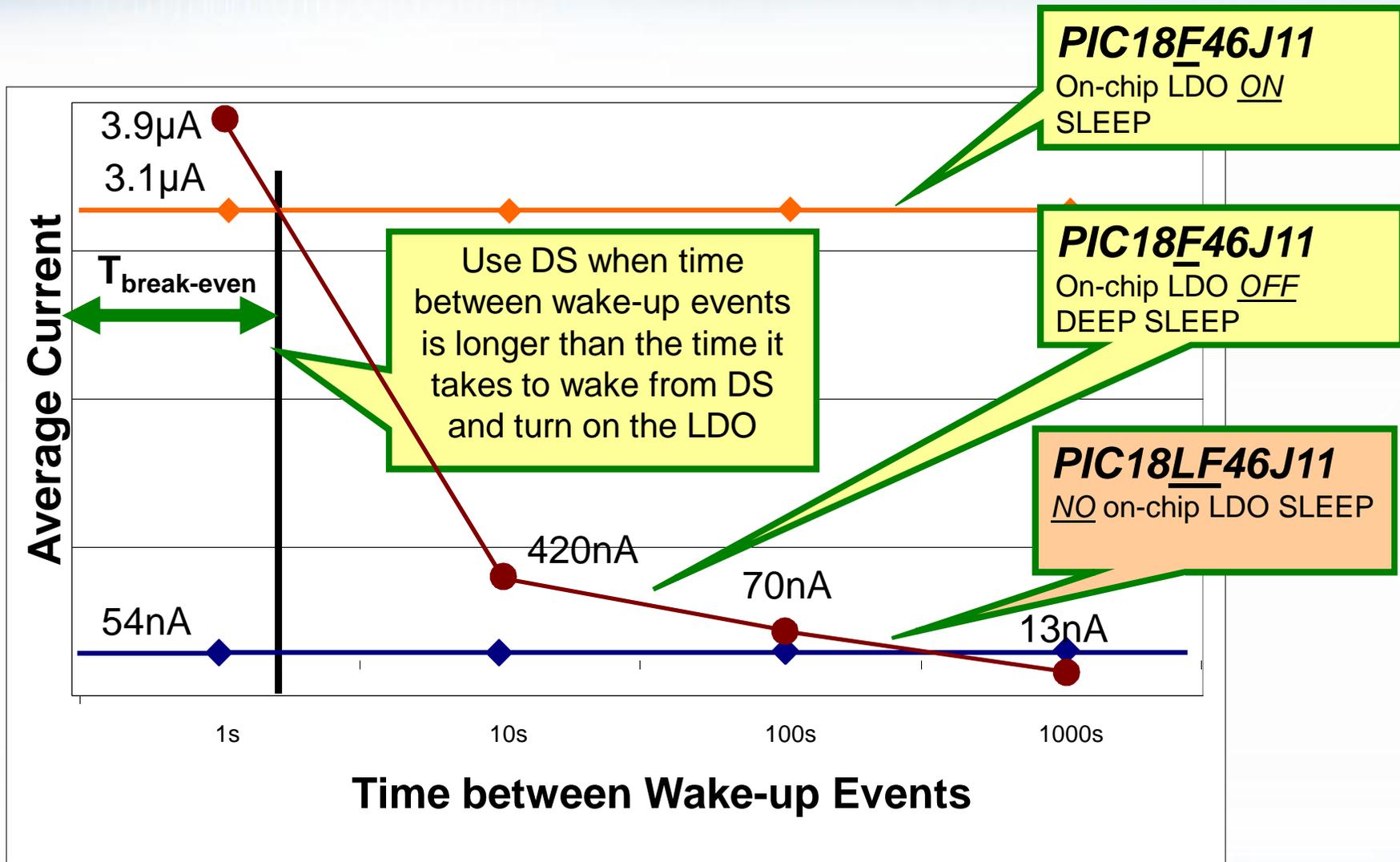
### Запуск с 2мя скоростями

- | Быстрый RC генератор стартует  $1\mu s - 5\mu s$  с % точностью
- | Обработываем приложение, пока стартует кварц
- | Переключаемся на кварц с PPM (миллионная доля) точностью, когда кварц готов



# Deep Sleep

## Когда использовать Deep Sleep?





# Deep Sleep

## Управляющие регистры DS

- | **Режим DS сохраняет питание этих регистров:**
  - | DSCON
    - | **Управляющий регистр DS**
  - | DSWSRC
    - | **DS Wake-Up Source Registers**
  - | DSGPR0 & DSGPR1
    - | **Регистры общего назначения для сохранения контекста, статуса или информации в DS**
    - | **16 бит для PIC24**
    - | **8 бит PIC18**

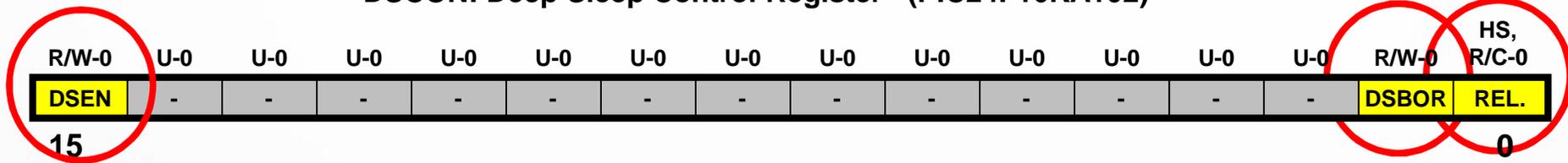
TABLE 4-21: DEEP SLEEP REGISTER MAP

File Name	Addr	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	All Resets <sup>(1)</sup>
DSCON	0758	DSEN	—	—	—	—	—	—	—	—	—	—	—	—	—	DSBOR	RELEASE	0000
DSWSRC	075A	—	—	—	—	—	—	—	DSINT0	DSFLT	—	—	DSWDT	DSRTCC	DSMCLR	—	DSPOR	0000
DSGPR0	075C	Deep Sleep General Purpose Register 0																0000
DSGPR1	075E	Deep Sleep General Purpose Register 1																0000



# Deep Sleep Управляющий регистр DS

DSCON: Deep Sleep Control Register (PIC24F16KA102)



- <15> **DSEN: Deep Sleep Enable Bit**  
 1 = enters Deep Sleep on PWRSV #0  
 0 = enters Normal Sleep on PWRSV #0
- <14:2> **Unimplemented: Read as '0'**
- <1> **DSBOR: Deep Sleep BOR Event bit**  
 1 = the DSBOR was active and a BOR event was detected during Deep Sleep  
 0 = the DSBOR was not active, or was active but did not detect a BOR event during DS
- <0> **RELEASE: I/O Pin State Release bit**  
 1 = Upon waking from Deep Sleep, I/O pins maintain their states previous to DS entry  
 0 = release I/O pins from their states previous to DS entry and allow TRIS & LAT bits to control their states

<b>Legend:</b>	C = Clearable bit	HS = Hardware Settable bit
R = Readable bit	W = Writable bit	U = Unimplemented bit, read as '0'
-n = Value at POR	'1' = Bit is set	'0' = Bit is cleared    x = Bit is unknown



# Deep Sleep

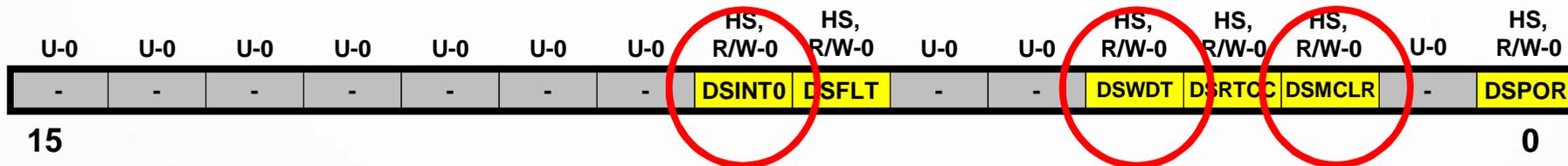
## *DSCON<RELEASE> Behavior*

- | **Upon POR (Power-On Reset) or wake-up from DS**
  - | I/O port TRIS & LAT bits are reset to default
  - | OSCCON<SOSCEN> bit is cleared
    - | **Disables SOSC (Secondary Oscillator)**
    - | **Has the effect of disabling RTCC**
  - | Core holds TRIS, LAT & SOSCEN control in state they were in before entering DS mode
    - | **Core does this by setting DSCON<RELEASE>**
- | **Application must:**
  - | Reconfigure TRIS, LAT, & SOSCEN
  - | Clear DSCON<RELEASE>
    - | **Turns I/O and SOSC control back over to application**
  - | Restore RAM and context information



# Deep Sleep DS Wake Up Sources

DSWSRC: Deep Sleep Wake Up Source Register (PIC24F16KA102)



<15:9> Unimplemented: Read as '0'

<8> DSINT0: Interrupt on change bit  
 1 = interrupt on change was asserted during DS  
 0 = interrupt on change was not asserted during DS

<7> DSFLT: DS Fault detected bit  
 1 = a fault occurred during DS and some DS config settings may have been corrupted  
 0 = no fault was detected during DS

<6:5> Unimplemented: Read as '0'

<4> DSWDT: DS watch dog timer time out bit  
 1 = the DS WDT timed out during DS  
 0 = the DS WDT did not time out during DS

<3> DSRTCC: DS Real Time Clock Calendar alarm bit  
 1 = the DSRTCC triggered an alarm during DS  
 0 = the DSRTCC did not trigger an alarm during DS

<2> DSMCLR: MCLR event bit  
 1 = the MCLR pin was active and asserted during DS  
 0 = the MCLR pin was not active or was active but not asserted during DS

<1> Unimplemented: Read as '0'

<0> DSPOR: Power On Reset bit  
 1 = the Vdd supply POR circuit was active and a POR event was detected  
 0 = the Vdd supply POR circuit was not active, or was active but did not detect a POR event



# Deep Sleep

## Просыпание из Deep Sleep

- | **DS сохраняет порты I/O и регистры DS**
- | **Возможность проснуться из DS по**
  - | INT0 (Interrupt 0)
  - | DSWDT (Deep Sleep Watchdog Timer)
  - | RTCC Alarm
  - | ULPWU (Ultra Low-Power Wake-Up)
- | **Пробудит из DS, но с ограничениями**
  - | Master Clear pin (MCLR)
    - | **Состояние портов I/O сбрасывается, потому что DSCON<RELEASE> немедленно очищается**
  - | Сброс питания
    - | **Состояние портов I/O сбрасывается**
    - | **Регистры Deep Sleep не сохраняются**



*Microchip's Annual*  
**MASTERS**

# ИТОГИ И ССЫЛКИ



# nanoWatt XLP

## Extreme Low Power Microcontrollers

PIC® MCU Family	Min. V <sub>dd</sub>	Family Members	Flash kB	Pins	Sleep (nA)	DS (nA)	WDT DSWDT (nA)	TMR1 RTCC (nA)	1MHz Run (µA)
PIC12LF1822	1.8	1	3.5	8	20	-	500	600	75
PIC16LF1827	1.8	5	3.5-7	18-28	20	-	500	600	80
PIC16LF1937 [LCD]	1.8	8	7-28	28-44	60	-	500	600	93
PIC16LF727	1.8	5	3.5-14	28-44	20	-	500	600	80
PIC18LF14K22	1.8	2	8-16	20	34	-	460	650	131
PIC18LF14K50 	1.8	2	8-16	20	24	-	450	790	125
PIC18F46J11	2.0	6	16-64	28-44	54	13	820	850	275
PIC18F46J50 	2.0	6	16-64	28-44	60	15	780	830	275
PIC18F46K20	1.8	8	8-64	28-44	100	-	600	600	131
PIC24F04KA201	1.8	2	4	14-20	25	20	400	500	195
PIC24F16KA102	1.8	4	8-16	20-28	25	20	400	500	195
PIC24FJ64GA104	2.0	4	32-64	28-44	200	20	200	500	250
PIC24FJ64GB004 	2.0	4	32-64	28-44	200	20	200	500	250

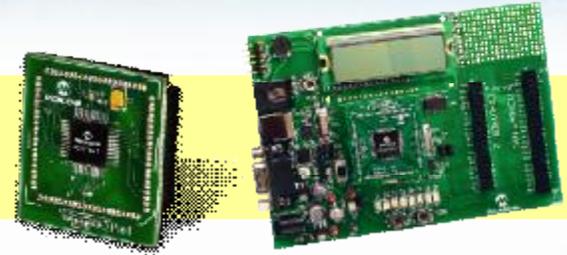


Current specifications are typical (TYP) values at minimum V<sub>dd</sub>

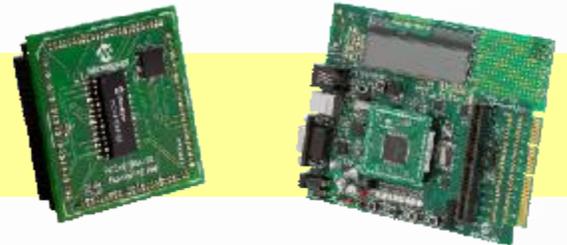


# ИТОГИ *nanoWatt XLP Tools*

PIC18 Explorer Board (DM183032)  
PIC18F46J11 PIM (MA180023)



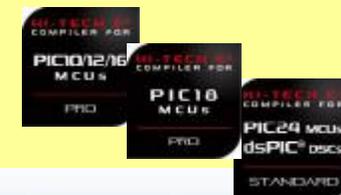
Explorer 16 Board (DM240001)  
PIC24F16KA102 PIM (MA240017)



XLP 16-bit Development Board (DM240311)



C Compilers  
Available from Microchip & HI-TECH  
Available in free evaluation versions





# Итоги References

**MICROCHIP** **AN879**

**Using the Microchip Ultra Low-Power Wake-Up Module**

Authors: Ruan Lourens  
Jose Henevedas

current consumption. These types of applications require a low-power periodic wake-up and can be accomplished by activating a low-power timer prior to

**MICROCHIP** **PIC24F Family Reference Manual**

**Section 39. Power-Saving Features with Deep Sleep**

**HIGHLIGHTS**

This section of the manual contains the following major topics:

39.1 Introduction	39-2
39.2 Microcontroller Clock Manipulation	39-7
39.3 Interrupt-Resilient Power-Saving Modes	39-5
39.4 Selection-Resilient Power Control	39-17
39.5 Debug Tips	39-20
39.6 Related Application Notes	39-21
39.7 Revision History	39-22

**MICROCHIP** **AN1267**

**nanoWatt and nanoWatt XLP™ Technologies:  
An Introduction to Microchip's Low-Power Devices**

Author: David Nay  
Microchip Technology Inc.

For many PIC devices, it also includes the clocking of logic necessary to resume operation from the Stand-By mode.

**MICROCHIP** **AN1288**

**Design Practices for Low-Power External Oscillators**

Author: Jonathan Aiken  
Microchip Technology Inc.

**PROBING THE CIRCUIT**

Oscillator circuits are highly sensitive to capacitance; therefore, special care needs to be taken when examining signals. A regular oscilloscope probe has 10-12 pF of capacitance, which can be utilized to stop oscillations. It is recommended that low capacitance probes be used, preferably with a JFET input, and that the OSC2 pin be probed instead of OSC1.

Many new devices incorporate Automatic Current Control (ACC) for the crystal oscillator drive circuit, where, to conserve power, the amplitude of the signal is reduced when the circuit is operating as intended. When examining the waveforms, this needs to be considered.

**INTRODUCTION**

Many Microchip microcontrollers have internal circuitry to drive a 32.768 kHz external crystal to provide an asynchronous clock signal to the Timer1 internal counter. Timer1 is a 16 bit counter which can be used to create a Real Time Clock (RTC) with a precision, 1 second overflow interrupt for system timing.

**CHAPTER 2  
PIC® Microcontroller Low Power  
Tips 'n Tricks**

**Table Of Contents**

<b>GENERAL LOW POWER TIPS 'N TRICKS</b>	
TIP #1 Switching Of External Circuits	2-2
TIP #2 Duty Cycle	2-2
TIP #3 Power Budgeting	2-3
TIP #3 Configuring Port Pins	2-4
TIP #4 Use High-Value Pull-Up Resistors	2-4
TIP #5 Reduce Operating Voltage	2-4
TIP #6 Use an External Source for VPL Core Voltage	2-6
TIP #7 Battery Backup for PIC MCUs	2-6
<b>DYNAMIC OPERATION TIPS 'N TRICKS</b>	

**TIPS 'N TRICKS INTRODUCTION**

Microchip continues to provide innovative products that are smaller, faster, easier to use and more reliable. The Flash-based PIC® microcontrollers (MCUs) are used in an wide range of everyday products, from simple capacitors, hospital ID tags and pet containment systems, to industrial, automotive and medical products.

PIC MCUs featuring nanoWatt technology implement a variety of important features which have become standard in PIC microcontrollers. Since the release of nanoWatt technology, changes in MCU process technology and



# ИТОГИ XLP Home Page

[www.microchip.com/xlp](http://www.microchip.com/xlp)

eXtreme Low Power - Windows Internet Explorer

www.microchip.com/xlp

File Edit View Favorites Tools Help

myMicrochip Login | English

Home Products Design Support Applications Buy/Sample Corporate What's New

**MICROCHIP**

search

**eXtreme Low Power**

Overview XLP Technology Featured Products Training & Support Getting Started

**PIC® MCUs with XLP are Battery Friendly**

**Lowest power sleep modes with flexible wake-up sources**

- ▶ Sleep currents as low as 20 nA
- ▶ Wake-up sources in every sleep mode
- ▶ Special low power BOR, WDT, RTC

**Longest battery life with robustness & reliability features**

- ▶ Enable battery lifetime > 20 years
- ▶ Maximum battery life for alkaline batteries and lithium coin cells
- ▶ Higher performance across 1.8 – 3V than MSP430

**Most efficient performance across battery voltage range**

- ▶ 80% more single cycle instructions
- ▶ Faster execution allows you to sleep longer
- ▶ High performance at low battery voltages

**Lowest Sleep Currents**

Device	Deep Sleep	Idle	Wait	Standby
MSP430F2252	~100 nA	~100 nA	~100 nA	~100 nA
PIC24F16KA102	~20 nA	~20 nA	~20 nA	~20 nA

**Maximize Battery Life with XLP**

Device	Battery Life (Years)
PIC24F16KA102	> 20
Competitor 1	~1000 days
Competitor A	~800 days

**Instruction Set Efficiency**

Device	Single Cycle Instructions
PIC	~80% more
MSP430	~0%

**New!**

- ▶ XLP 16-bit Dev Board available
- ▶ XLP Battery Life Estimator

**Videos**

- ▶ XLP Intro (2m13s)
- ▶ Compare PIC24 XLP to ULP (5m06s)
- ▶ PIC16 Low Power Benchmarks (3m01s)

**Find XLP Products**

- ▶ Product Selection Tool

Go to a product:

**Resources**

- ▶ XLP Battery Life Estimator
- ▶ nanoWatt XLP Technology: An Introduction to Microchip's Low-Power Devices (AN1267)
- ▶ Practical Applications of Low-Power Design with nanoWatt XLP
- ▶ PIC® MCU Time-to-Market Features



# Want More?

[www.microchip.com/microsolutions](http://www.microchip.com/microsolutions)



## microSOLUTIONS MAR 2011

### In This Issue...

1. Microchip Sets New Benchmark for Low-Power Microcontrollers; Significantly Expands Enhanced 8-bit PIC® MCU Portfolio
2. Synchronous Boost Regulator from Microchip Enables Longer-Lasting Battery Applications
3. Product Highlight: PIC32 with Ethernet
4. Join the Microchip Community on Your Favorite Websites
5. Non-Volatile, Quad-Digital, Programmable On-Chip State Element Capabilities of 8-Microcontrollers (MCU)
6. Looking for More RAM?
7. USB-to-UART Protocol Converter Makes it Easy to Add USB to Existing Systems
8. Need Low-Power Analog?
9. Implementing Ethernet with the mTCP™ Central Node Controller
10. Enhance Your Core Design Software on PIC18 SoC Kits
11. Interact with Microchip at MCHP™ Trade Shows
12. Looking to Enhance Your Embedded Control Designs?
13. Microchip Around Town: ESC SV 2010 - FREE TRAINING
14. Microchip Around Town: ESC SV 2010 - SPEAKING SOLUTIONS
15. Microchip Around Town: ESC SV 2010 - SHARING SOLUTIONS
16. New to microchipDIRECT?
17. What's New in Microchip Literature?



### Microchip Sets New Benchmark for Low-Power Microcontrollers; Significantly Expands Enhanced 8-bit PIC® MCU Portfolio

New MCUs Feature Less Than 50 µA/MHz Active Current; Industry-Leading Peripheral Integration



Microchip Technology Inc. unveiled several new 8-bit PIC® microcontrollers (MCUs) that sets the industry benchmark for low-power microcontrollers and peripheral integration. These new MCUs feature active currents of less than 50 µA/MHz and sleep currents down to 20 nA. The PIC12F182X MCUs extend Microchip's Enhanced Mid-range 8-bit core product line into the 8-pin segment, and include mTouch™ capacitive touch-sensing, and communications peripherals. The PIC16F19XX MCUs feature a broad range of peripherals, such as mTouch capacitive touch-sensing module, LCD drive, multiple communications and more Pulse Width Modulator (PWM) peripherals. All of these general-purpose MCUs are well suited for applications in the appliance, consumer, industrial and automotive markets, among others.

nanoWatt XLP technology remains the standard for battery-friendly MCUs, which, combined with the extremely low active current consumption of these new MCUs, improves overall energy efficiency to levels currently not readily available. Microchip's Enhanced Mid-range 8-bit architecture provides an up to 50% increase in performance, and 14 new instructions that result in up to 40% better code execution over previous-generation 8-bit PIC16 MCUs. The PIC1XF182X MCUs include dual I²C™/SPI interfaces, multiple PWM channels with independent time bases, a Data Signal Modulator and other peripherals that enable designers to combine many functions into a single MCU. The PIC16F19XX MCUs provide up to 28 KB of Flash program memory and numerous enhanced capabilities. The on-chip LCD drive supports up to 184 segments and provides a low-power drive mode for increased efficiency. The MCUs also include up to 5 PWM channels with independent time bases for controlling various motor types and peripherals.

Designers can use Microchip's F1 Evaluation Platform (part # DM164130-1, \$39.99) for developing with Enhanced 8-bit PIC MCUs. The platform includes a 44-pin development board populated with a PIC16LF1937 MCU, prototyping space, 3V LCD glass, support for the PICKit™ 3 In-Circuit Debugger/Programmer (part # PG164130, \$44.95) and a motor control add on. The PIC16F1937 Plug-In Module (part # MA160012, \$25) for Microchip's PIC18 Explorer Board (part # DM183032, \$99.99) is also available.

Microchip's PICDEM™ Lab Development Kit (part # DM163035, \$124.99) can be used with the PIC1XF182X MCUs. The kit comes complete with a development board containing five popular 8-bit PIC MCUs, a bag of discrete components, a debugger/programmer and a CD containing a User's Guide, labs and application examples.

The PIC1XF182X and PIC16F19XX MCUs are available in DFN, PDIP QFN, SOIC, TQFP TSSOP and UQFN packages of varying sizes, from 8- to 64-pins; at prices ranging from \$0.69 to \$1.74 each, in 10,000-unit quantities.

For more information, visit: <http://www.microchip.com/Enhanced>

[www.microchip.com](http://www.microchip.com)

Microcontrollers • Digital Signal Controllers • Analog • Serial EEPROMs



*Microchip's Annual*  
**MASTERS**

# Практическая часть





# Практическая часть Обзор демоплаты

JP12: переключение режимов питания микроконтроллера

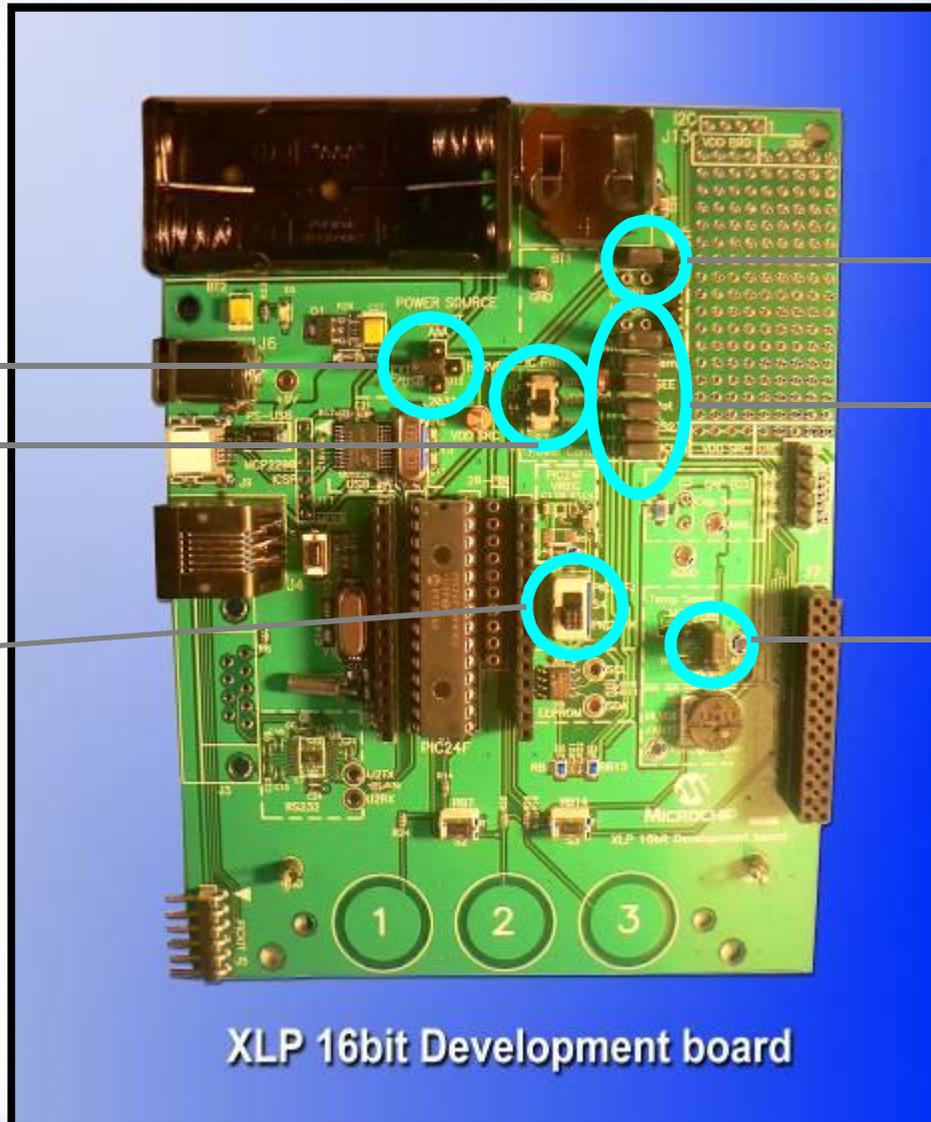
S7: переключение режимов питания периферии

S4: переключение типа микроконтроллера

JP9: цепь питания микроконтроллера

JP10,1,2,3,7,6: включение/отключение питания периферии

JP5: переключение типа датчика температуры



XLP 16bit Development board



# Практическая часть Предустановка

JP12: в положение  
EXT PS/USB

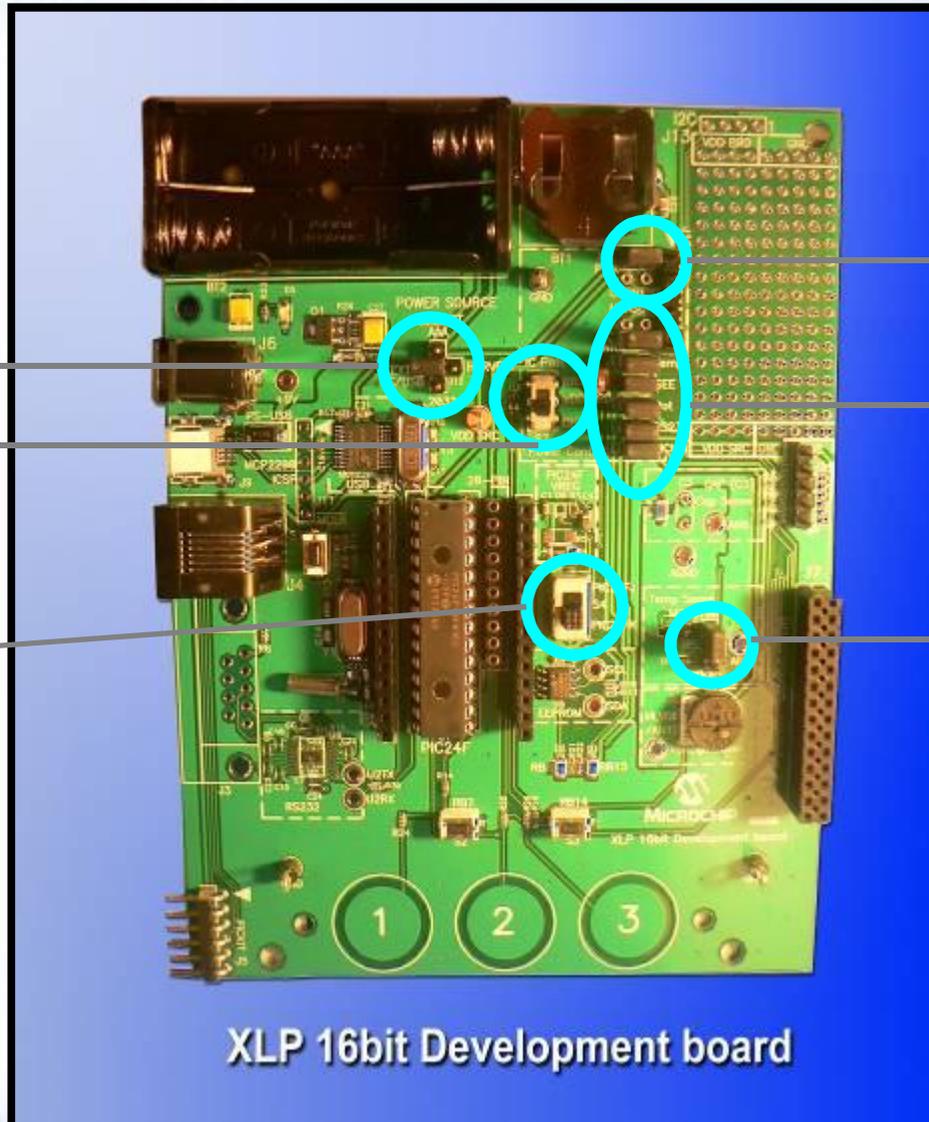
S7: в положение  
ON

S4: в положение  
PIC24FK

JP9: включено

JP10,1,2,3,7,6:  
включены

JP5: в положение  
MCP9700



XLP 16bit Development board



# Lab Exercise #3

## Сборка или «что нам понадобится»

- | **DM240311 16-Bit XLP Development Board**
- | **Набор резисторов**
- | **Мультиметр**
- | **Программатор/отладчик**
- | **USB кабель:**
  - | Питание
  - | Подключение к ПК через мост USB-UART

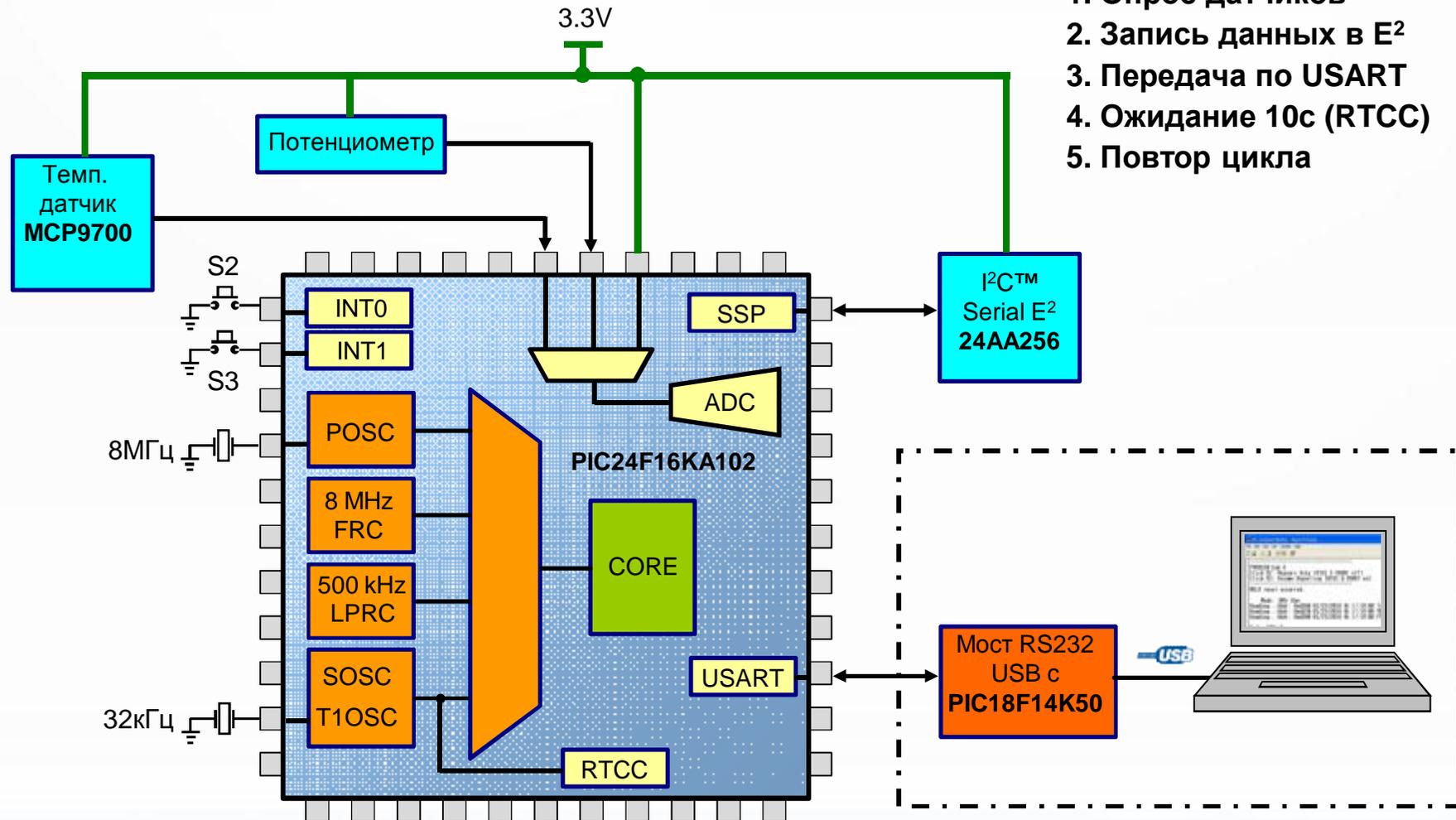




# Практическая часть Блок-схема или «что мы собрали»

## Цикл работы

1. Опрос датчиков
2. Запись данных в E<sup>2</sup>
3. Передача по USART
4. Ожидание 10с (RTCC)
5. Повтор цикла



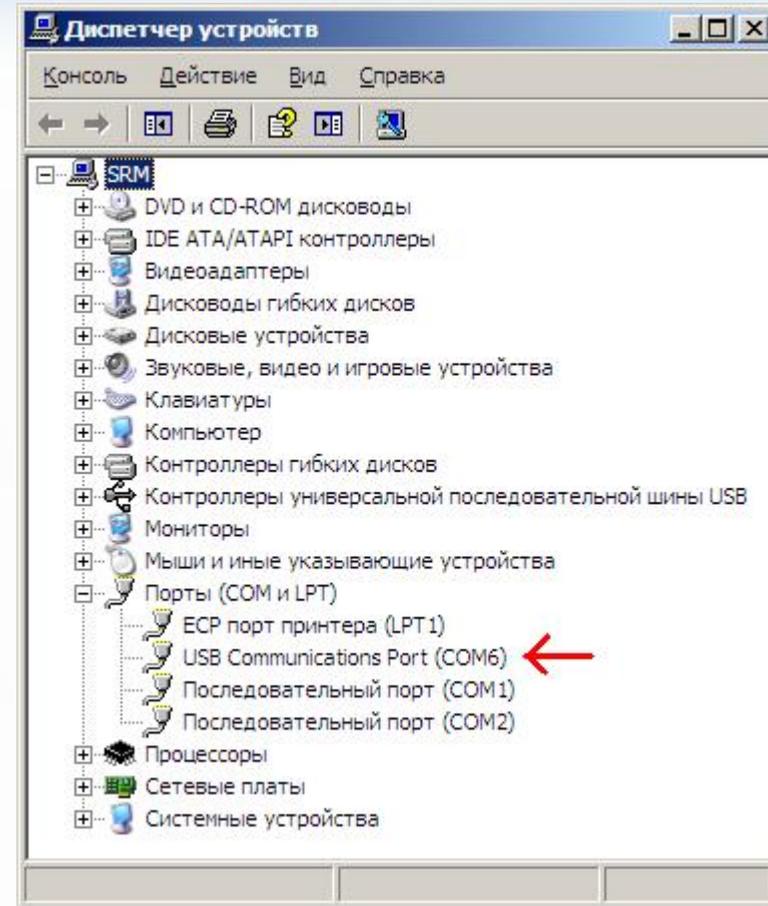


# Практическая часть

## Подключение к ПК

После подключения по USB демоплата использует один из свободных COM-портов. Какой именно, можно посмотреть в настройках Windows:

Пуск->Настройка->Панель управления->Система->Оборудование->Диспетчер устройств

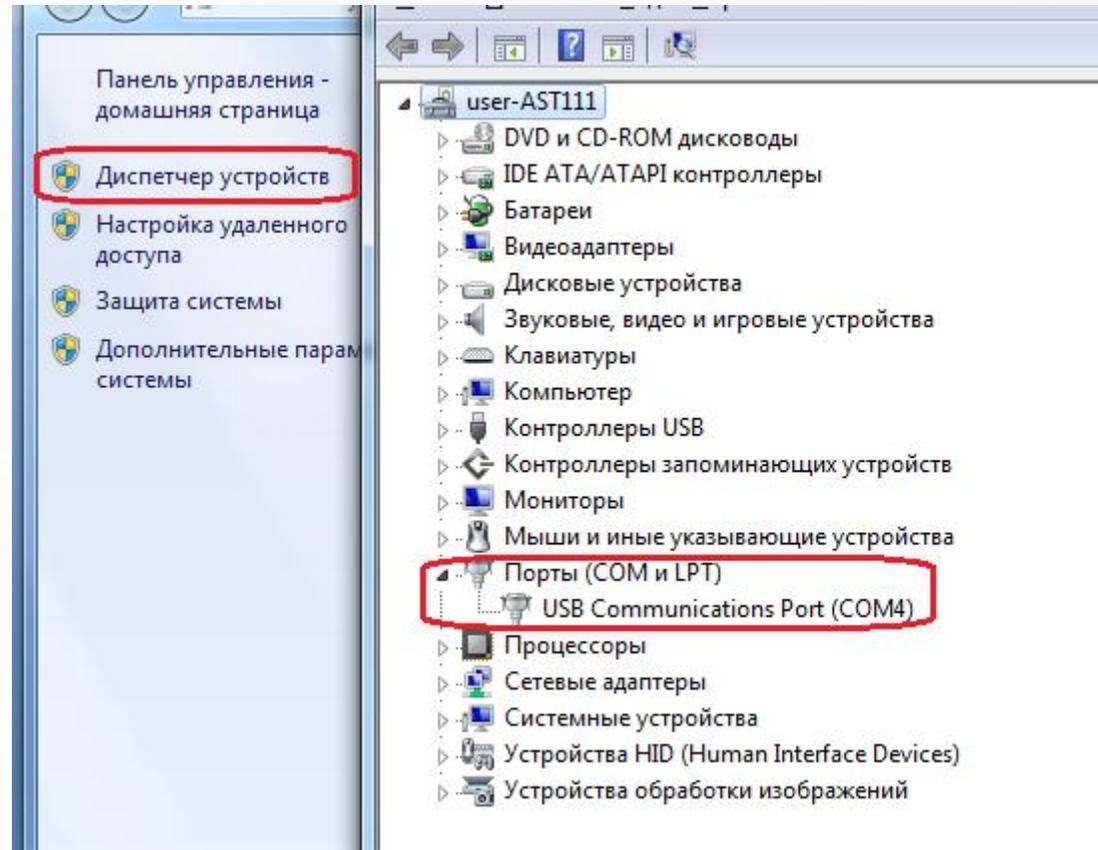




# Практическая часть

## Подключение к ПК

Или вариант для  
Windows 7:  
Пуск->Панель  
управления->Система,  
Далее - «Диспетчер  
устройств».





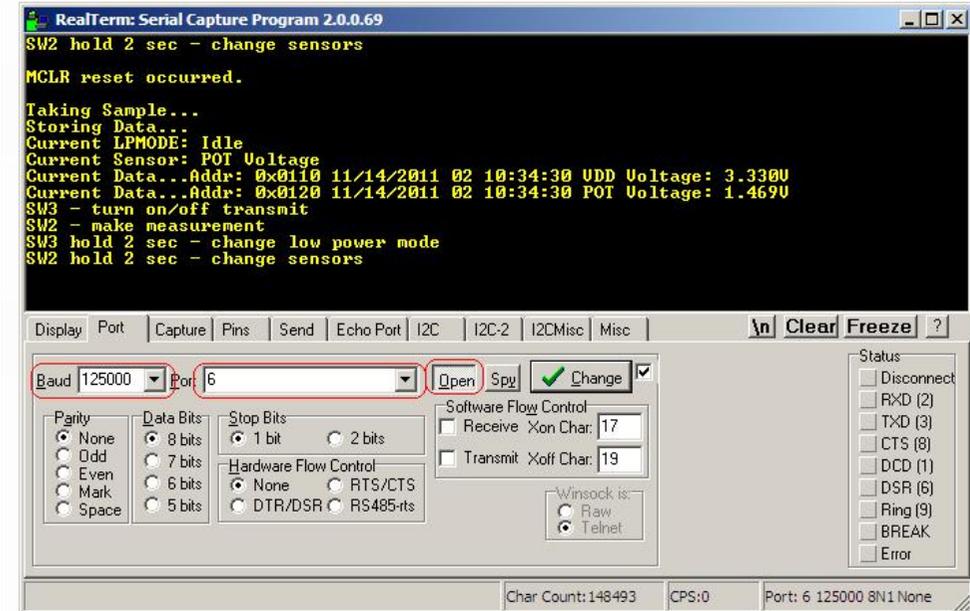
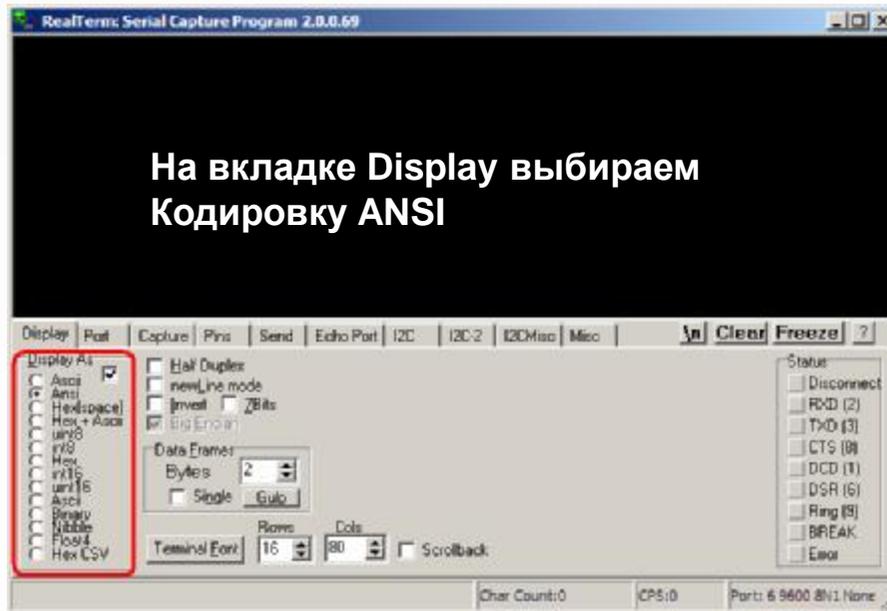
# Практическая часть Подключение к ПК

Теперь настраиваем программу-терминал. Запускаем Realterm:



На вкладке Port выбираем нужный порт, Выставляем Baud=125000 и нажимаем кнопку Open. После нажатия S1 на плате будет переданы строчки с информацией.

На вкладке Display выбираем Кодировку ANSI





# Практическая часть

## Подключение к ПК

Порядок обработки нажатий кнопок следующий:

**S1 – Master Clear микроконтроллера;**

**S2 – замер показаний датчиков и вывод на экран;**

**S3 – вкл/выкл вывода на экран;**

**S2 зажата в течение 2 сек – переключение активных датчиков;**

**S3 зажата в течение 2 сек – переключение режима низкого потребления;**

**S2 зажата при нажатии S1 – очистка EEPROM и RTCC;**

**S3 зажата при нажатии S1 – передача полного содержимого EEPROM.**

```
SW3 - turn on/off transmit  
SW2 - make measurement  
SW3 hold 2 sec - change low power mode  
SW2 hold 2 sec - change sensors
```



# Практическая часть

## Подключение к ПК

### Внимание!

**В режиме DeepSleep микроконтроллером обрабатывается только нажатие на S2 и сигнал будильника RTCC, поэтому вывести его из работы в режиме DeepSleep по нажатию кнопки S3 не получится.**

**Делаем так:**

- 1. Зажимаем S3;**
- 2. Нажимаем и отпускаем S2;**
- 3. По прошествии двух секунд отпускаем S3.**



# Практическая часть

## Работаем с проектом

Открываем MPLAB IDE и демо-проект: `C:\mastersXLP\source.`

Структура проекта:

- `buttons.c` – функции детектирования и обработки нажатий кнопок S1 и S2;
- `ctmu.c` – функции работы с емкостными кнопками;
- `eeprom.c` – функции работы с EEPROM на плате;
- `rtcc.c` – функции работы с часами реального времени;
- `sensors.c` – функции работы с датчиками;
- `transmit.c` – функции подготовки и отправки данных на ПК;
- `uart.c` – функции передачи данных по USART;
- `XLP16Demo.h` – основные константы работы платы.



# Практическая часть

## Работаем с проектом

### XLP16Demo.h.

```
#define USE_PWRCTL – управление питанием периферии;  
#define USE_CAPTOUCH – использование емкостных кнопок;  
#define DEFAULT_MODE – режим энергосбережения по умолчанию;  
#define DEFAULT_SENSOR – активные сенсоры по умолчанию;  
#define OSC_MODE – режим работы 8 МГц или 1 МГц;  
#define OPTIMIZE_A – уровень оптимизации потребления А;  
#define OPTIMIZE_B – уровень оптимизации потребления В;  
#define USE_RTCC_ALARM – использование будильника RTCC;  
#define ALARM_PERIOD – период срабатывания будильника RTCC.
```

### Настройки по умолчанию:

```
USE_PWRCTL – выключено;  
USE_CAPTOUCH – включено;  
DEFAULT_MODE - установлено LP_IDLE – режим Idle;  
DEFAULT_SENSOR – установлено MODE_TEMP – датчик температуры;  
OSC_MODE – установлено MHz_8 – 8 МГц;  
OPTIMIZE_A – выключено;  
OPTIMIZE_B – выключено;  
USE_RTCC_ALARM – включено;  
ALARM_PERIOD – установлено TEN_SECOND – 10 секунд.
```



# Практическая часть

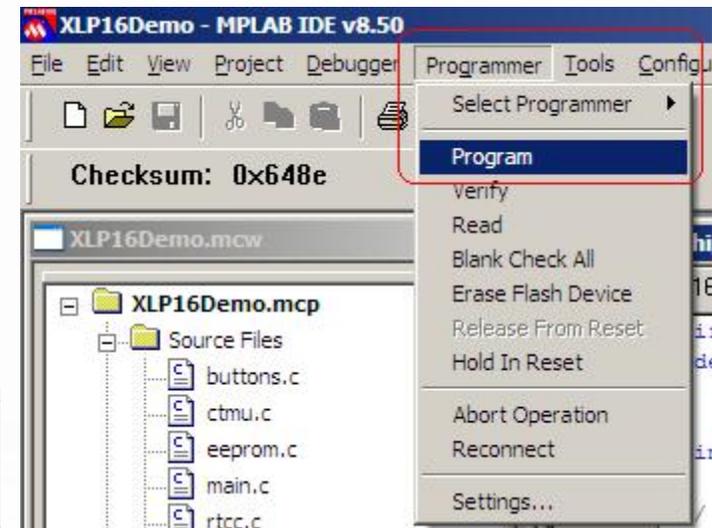
## Емкостные кнопки – хорошо, но DeepSleep лучше.

Отключаем ЕК, закомментировав опцию :

```
#define USE_CAPTOUCH
```

Пересобираем проект и программируем микроконтроллер.

```
#define OSC_8      // режим работы на 8 МГц  
// #define OSC1    // режим работы на 1 МГц  
  
// включить сенсорные кнопки  
#define USE_CAPTOUCH  
  
// режимы по умолчанию  
#define DEFAULT_MODE LP_IDLE
```





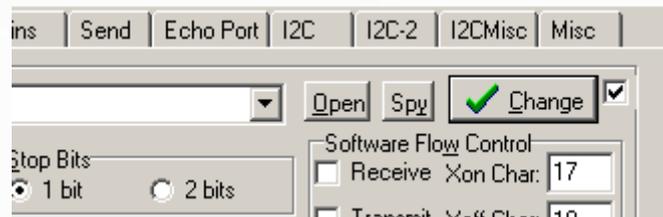
# Практическая часть

## Начинаем эксперименты

Основная цель лабораторной – исследовать потребляемый микроконтроллером ток в различных режимах.

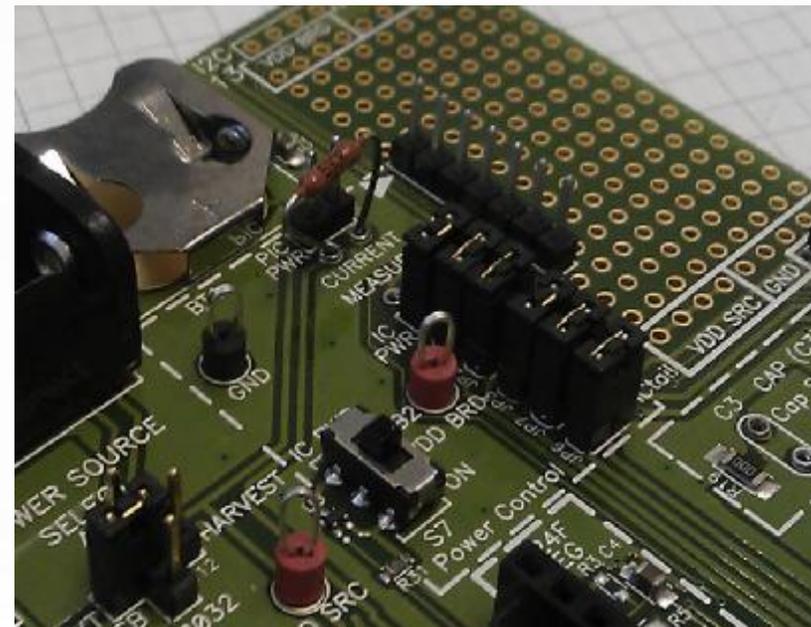
Попробуем.

1. Выключите работу терминала, нажав на кнопку “Open”:



2. Отключите USB-кабель от платы.

3. Отключите JP9 и вставьте резистор 560.Ом в отверстия рядом с JP9:





# Практическая часть Несколько уточнений

1. При отключении платы от USB перед повторным включением стоит отключить программу-терминал.

2. При использовании резистора слишком большого номинала контроллер может не выйти из режимов Sleep и DeepSleep из-за слишком большого падения напряжения в цепи питания. Здесь можно поступить так – замкнуть перемычку JP9, запустить контроллер, перевести его в режим Sleep или DeepSleep, установить резистор большого номинала, разомкнуть перемычку JP9.

Выполняется это для замера слишком маленьких токов.

3. После перепрограммирования микроконтроллера для того, чтобы увидеть в программе-терминале информацию от контроллера стоит нажать кнопку S1.

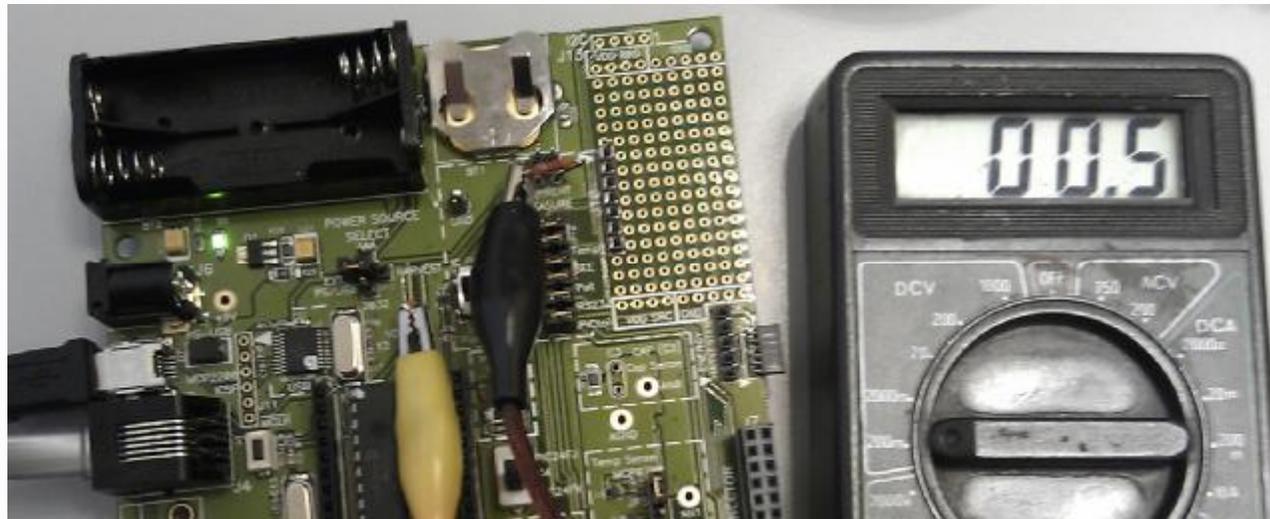
4. Если что-то не получается – не стоит торопиться.



# Практическая часть

## Начинаем эксперименты

Теперь снова подключаем USB, открываем программу-терминал, подключаем мультиметр в режиме измерения мВ так, чтобы видеть падение напряжения на резисторе – красным щупом на точку VDD SRC, черным – на ножку резистора.



Кнопкой S3 переводим контроллер последовательно в режимы Idle(режим по умолчанию), Sleep и DeepSleep. Оцениваем потребляемый ток, по закону Ома:

$$I_{порт} = U_{рез}/R_{рез}$$



# Практическая часть

## Сохраняем результаты

Снятые результаты переносим в таблицу:  
C:\mastersXLP\Результаты.xls

Уровень оптимизации	Режим работы	Падение напряжения [мВ]	Потребляемый ток [мкА]
USE_PWRCTL	Idle	2131	2 663,75000
	Sleep	32323	40 403,75000
	DeepSleep	332	415,00000
OPTIMIZE_A	Idle	22	27,50000
	Sleep	3232	4 040,00000
	DeepSleep	333	416,25000
OPTIMIZE_B	Idle	111	138,75000
	Sleep	123	153,75000
	DeepSleep	321	401,25000
	R [Ом]	800	

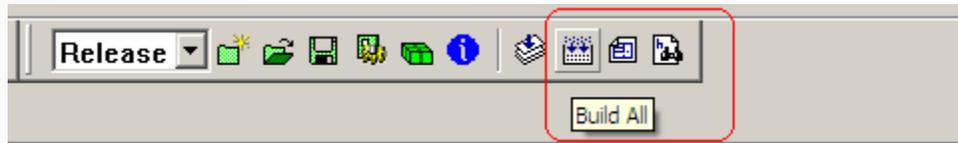
Вводим номинал резистора в Ом и падение напряжения в мВ, потребляемый ток считается автоматически по закону Ома.



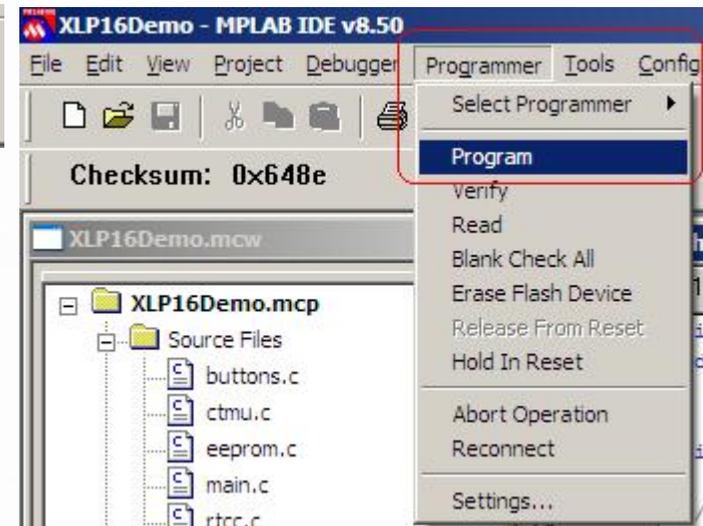
# Практическая часть

## Продолжаем эксперименты

Попробуем последовательно поменять настройки проекта. После изменений настроек потребуется компиляция и программирование контроллера.



Измените последовательно все настройки проекта, каждый раз компилируя проект и наблюдая, как изменится потребление контроллера.



```
#define OSC_8 // режим работы на 8 МГц
//#define OSC1 // режим работы на 1 МГц

//включить сенсорные кнопки
#ifdef OSC_8
// #define USE_CAPTOUCH
#endif
```



# Практическая часть

## Меняем настройки

1. Предварительно попробуем дать контроллеру возможность управлять питанием периферии. Для этого активируем опцию:

```
#define USE_PWRCTL
```

и меняем состояние переключателя S7 на «RB2». Пересобираем проект, программируем контроллер, замеряем потребление на разных режимах.

```
// управление питанием периферии
#define USE_PWRCTL
// уровень оптимизации
// #define OPTIMIZE_A           // выключение неиспользуемых
// #define OPTIMIZE_B           // дополнительное

#define OSC_8           // режим работы на 8 МГц
// #define OSC1           // режим работы на 1 МГц

// включить сенсорные кнопки
#ifdef OSC_8
// #define USE_CAPTOUCH
#endif
```

The screenshot shows the MPLAB IDE v8.50 interface. The 'Program' menu is open, showing options like 'Verify', 'Read', 'Blank Check All', 'Erase Flash Device', 'Release From Reset', 'Hold In Reset', 'Abort Operation', 'Reconnect', and 'Settings...'. The 'Build All' button in the toolbar is also highlighted with a red box.



# Практическая часть

## Меняем настройки

2. Включаем оптимизацию по уровню OPTIMIZE\_A:

```
#define OPTIMIZE_A
```

В этом режиме будет отключена вся неиспользуемая периферия. Компилируем проект и программируем микроконтроллер.

```
// управление питанием периферии
#define USE_PWRCTL
// уровень оптимизации
#define OPTIMIZE_A           // выкл
//#define OPTIMIZE_B         // до:

#define OSC_8           // режим работы н:
//#define OSC1         // режим работы н:
```

Замеряем падение напряжения на резисторе.



# Практическая часть

## Меняем настройки

3. Включаем оптимизацию по уровню OPTIMIZE\_V:

```
#define OPTIMIZE_V
```

В этом режиме будет отключаться периферия по окончании работы с ней. Так, например, АЦП будет отключаться по окончании замера.

```
// управление питанием периферии
#define USE_PWRCTL
// уровень оптимизации
#define OPTIMIZE_A          // выключение н
#define OPTIMIZE_V          // дополнительн

#define OSC_8              // режим работы на 8 МГц
//#define OSC1              // режим работы на 1 МГц

//включить сенсорные кнопки
#ifdef OSC_8
// #define USE_CAPTOUCH
#endif
```

Замеряем падение напряжения на резисторе.



# Практическая часть

## Мультиметр – штука тонкая

И вот мы столкнулись с проблемой – потребление в режимах Sleep и DeepSleep стало настолько низким, что падение напряжение на резисторе 560.Ом исчисляется микровольтами. Мультиметр, скорее всего, дает значение с погрешностью. Если попробовать поставить резистор побольше, то контроллер не сможет выйти из режимов Sleep и DeepSleep, так как при выходе возрастет потребляемый ток, а, значит, и падение напряжения в цепи питания.

Попробуем следующий вариант:

1. Отключите использование будильника RTCC – контроллер будет выходить из Sleep или DeepSleep только по нажатию кнопок.

Сделать это можно, отключив опцию:

```
#define USE_RTCC_ALARM
```

```
// использование будильника  
#define USE_RTCC_ALARM  
  
#define ALARM_PERIOD    TEN_SECOND  
#define ONE_SECOND     10  
#define TEN_SECOND     100
```



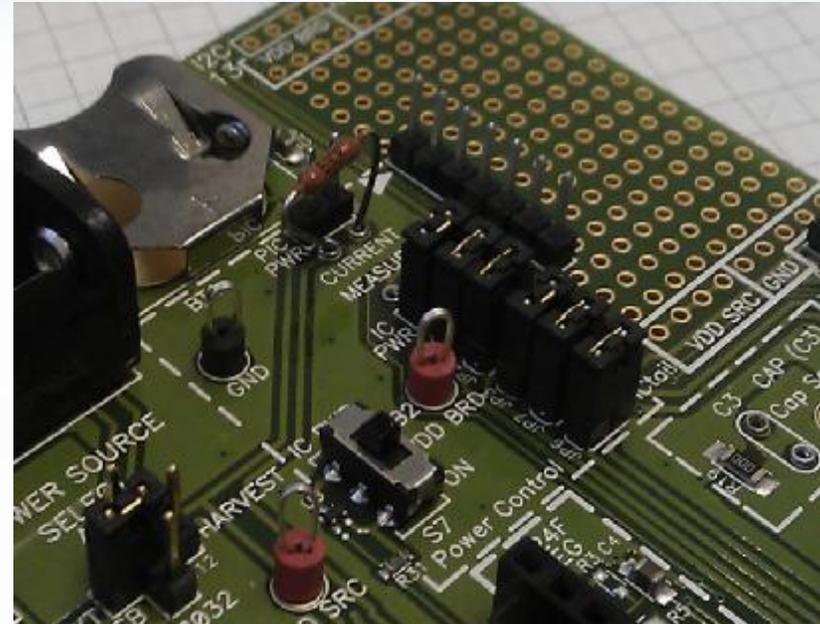
# Практическая часть

## Мультиметр – штука тонкая

2. Замкните перемычку JP9 и подключите плату к USB. Вместо резистора 560.Ом установите резистор на 130 кОм.

3. Переведите контроллер в режим Sleep или DeepSleep удерживанием S3 в течение 2-х секунд.

4. Разомкните перемычку JP9 – ток потечет через резистор. На столь большом резисторе падение напряжение будет уже ощутимо для мультиметра, и будет возможно замерить потребляемый ток.





# Практическая часть

## Меняем настройки

4. Меняем тактовую частоту с 8 МГц на 1 МГц:

```
///define OSC_8  
define OSC1
```

```
// управление питанием периферии  
define USE_PWRCTL  
// уровень оптимизации  
define OPTIMIZE_A           // выключение неиспользуемых периферийных устройств  
define OPTIMIZE_B           // дополнительные оптимизации  
  
///define OSC_8           // режим работы на 8 МГц  
define OSC1             // режим работы на 1 МГц
```

Замеряем падение напряжения на резисторе.



# Планирование потребления *nanoWatt XLP Battery Life Estimator*



## **Выбрать МК и параметры**

Новые МК добавляются с новой версией, или вы сами можете их добавлять.  
На пример: "PIC18LFxxJ11.csv"

## **Выбрать батарейку**

Новые батарейки могут добавляться изменением файла  
"CustomBattery.csv"

## **Ввести режимы и потребление**

## **Пересмотреть и сохранить**

Microchip XLP Battery Life Estimator

Step 1: Select Device  
PIC24F16KA102 Device Notes

Step 2: Select Parameters  
Voltage: 3.3V Temperature: 25

Step 3: Select Battery

Description	Capacity	Self-discharge	Nominal V	Max cont. I	Max pulse I	Notes
LiMnO2 (CR2032)	225 mAh	0.08 %/mo(0.25 uA)	3.0 V	1.8 mA	12 mA	*
Li(CF)n (BR2032)	190 mAh	0.04 %/mo(0.106 uA)	3.0 V	0.2 mA	2 mA	*
LiFeS2 (AAA)	1150 mAh	0.05 %/mo(0.799 uA)	1.5 V	750 mA	750 mA	*
LiFeS2 (AA)	3000 mAh	0.05 %/mo(2.083 uA)	1.5 V	2000 mA	2000 mA	*
Li-ion (Approx. 'A')	2250 mAh	5 %/mo(0.156 mA)	3.6 V	2250 mA	2250 mA	*
Alkaline (AAA)	1150 mAh	0.3 %/mo(4.792 uA)	1.5 V	*	*	Drain rate affe
Alkaline (AA)	2850 mAh	0.3 %/mo(11.875 uA)	1.5 V	*	*	Drain rate affe

Step 4: Add/Modify Operational Mode

Label	Mode	Frequency	Time	Current	Charge	Peripherals
SLEEP w/RTCC	DEEP SLEEP	*	10 S/100%	0.835 uA	2.319 nAh	RTCC (w/ SOSC)

Estimated Battery Life: 23 years 245 days 13 hours

Estimated Average Current: 0.835 uA

Estimated Peak Current: 0.835 uA

Buttons: Load Settings, Save Settings, Create Report, Quit



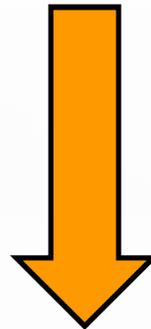
# Планирование потребления *nanoWatt XLP Battery Life Estimator*



**800мкс - 6 мкА - опрос температуры;**

**800мкс - 6 мкА - опрос потенциометра;**

**5 мс - 3 мА – запись в EEPROM.**





# Планирование потребления *nanoWatt XLP Battery Life Estimator*



Microchip XLP Battery Life Estimator

Step 1: Select Device  
PIC24F16KA102

Step 2: Select Parameters  
Voltage: 3.3V

Step 3: Select Battery

Description	Capacity	Self-discharge	Nominal V	Max cont. I	Max
LiMnO2 (CR2032)	225 mAh	0.08 %/mo(0.25 uA)	3.0 V	1.8 mA	12
Li(CF)n (BR2032)	190 mAh	0.04 %/mo(0.106 uA)	3.0 V	0.2 mA	2
LiFeS2 (AAA)	1150 mAh	0.05 %/mo(0.799 uA)	1.5 V	750 mA	75
LiFeS2 (AA)	3000 mAh	0.05 %/mo(2.083 uA)	1.5 V	2000 mA	20
Li-ion (Approx. 'A')	2250 mAh	5 %/mo(0.156 mA)	3.6 V	2250 mA	22
Alkaline (AAA)	1150 mAh	0.3 %/mo(4.792 uA)	1.5 V	*	*
Alkaline (AA)	2850 mAh	0.3 %/mo(11.875 uA)	1.5 V	*	*

Step 4: Add/Modify Operational Mode

Label	Mode	Frequency	Time	Current
RUN A - READ MCP9700	RUN	8 MHz (FRC)	0.8 mS/100%	3.057 mA

Buttons: Add, Modify, Remove

Estimated Battery Life: 3 days 1 hour

Estimated Average Current: 3.057 mA (> cont. battery current)

Buttons: Load Settings, Save Settings, Create Report, Quit

Add/Modify mode dialog:

Mode Settings

Duration: 0.8 mS

Additional System Current (optional): 6 uA

Name (optional): RUN A - READ MCP9700

PIC Microcontroller Settings

PIC Operating Mode: RUN

Clock Frequency: 8 MHz (FRC)

Peripherals:

- BOR <less than 4.5 uA>
- HLVD <less than 6.2 uA>
- LPBOR <95 nA>
- RTCC (w/ SOSC) <0.8 uA>
- Timer1 (w/ SOSC) <0.73 uA>
- WDT <0.87 uA>

Power Consumption

Microcontroller Charge: 0.678 nAh

Microcontroller Current: 3.051 mA

Total Charge (uC + System): 0.679 nAh

Total Current (uC + System): 3.057 mA

Buttons: OK, Cancel



*Microchip's Annual*  
**MASTERS**

**И на этом все.  
Спасибо за внимание!**

