



# **MICROCHIP** *2010*

---

## *MASTERS Conference*

# WIL

**Разработка беспроводных сетей при  
помощи решений Microchip**

# Цели класса

- q **По окончании данного класса Вы:**
  - q Узнаете о решениях, предлагаемых Microchip для построения беспроводных сетей
  - q Изучите основные протоколы, применяемые в беспроводных сетях
  - q Узнаете об отладочных средствах, предлагаемых Microchip для разработки беспроводных приложений
  - q Попробуете создавать беспроводные приложения самостоятельно

# Решаемые задачи

- q Необходимо разработать новое изделие, обеспечивающее беспроводной обмен данными с другими изделиями
- q Необходимо заменить проводную связь в уже разработанных устройствах на беспроводную

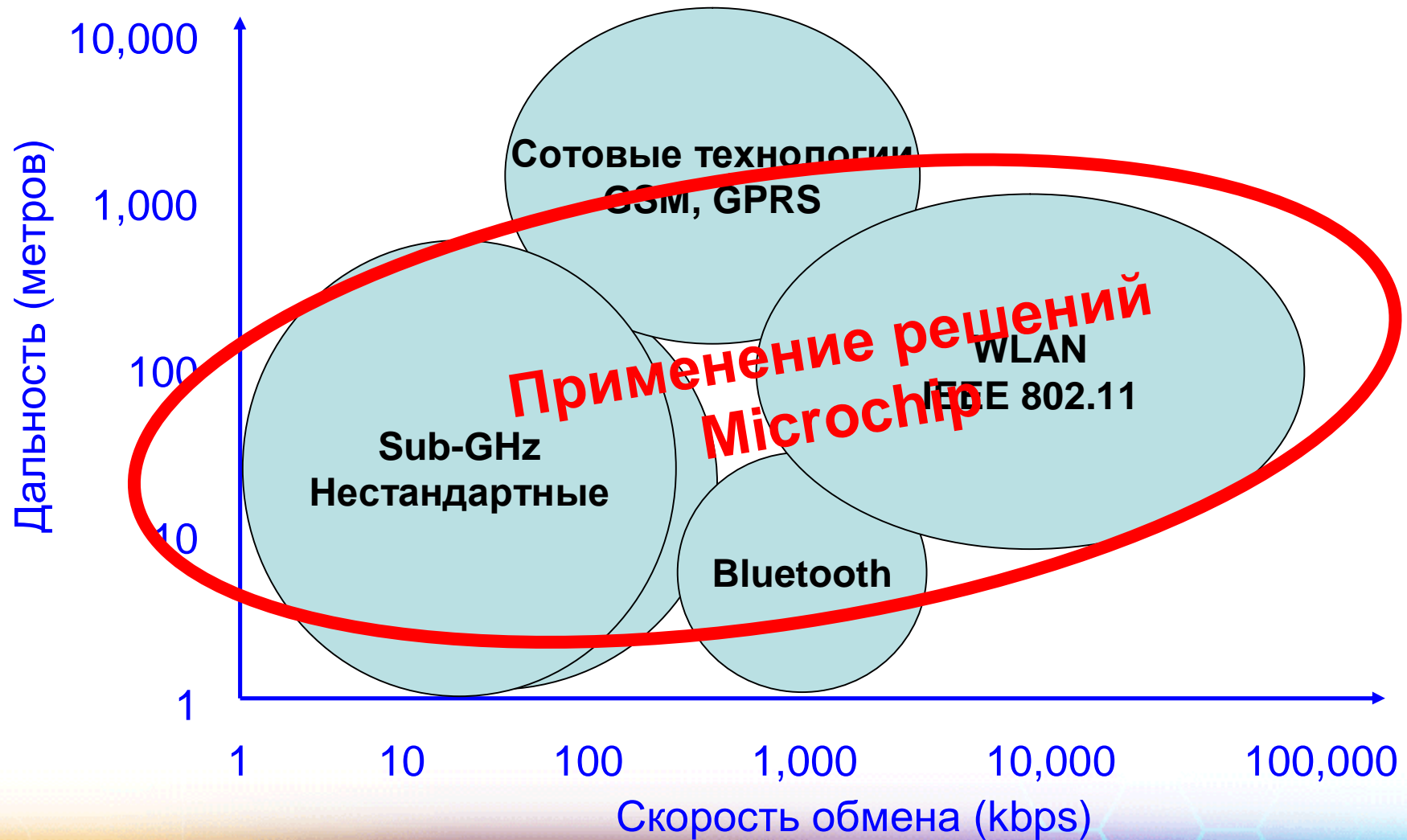


**Время на разработку ограничено**

# Выбор беспроводного решения

- **Дальность / частотный диапазон**
- **Скорость / объем передаваемых данных**
- **Потребляемый ток / источник питания**
- **Совместимость / стандартизация**
- **Топология / количество узлов**
- **Имеющиеся ресурсы**

# Беспроводные протоколы



# Модель OSI

Уровни передачи данных

Прикладной уровень

Уровень представления

Сеансовый уровень

Транспортный уровень

Сетевой уровень

Канальный уровень

Физический уровень

Уровни доступа к среде передачи

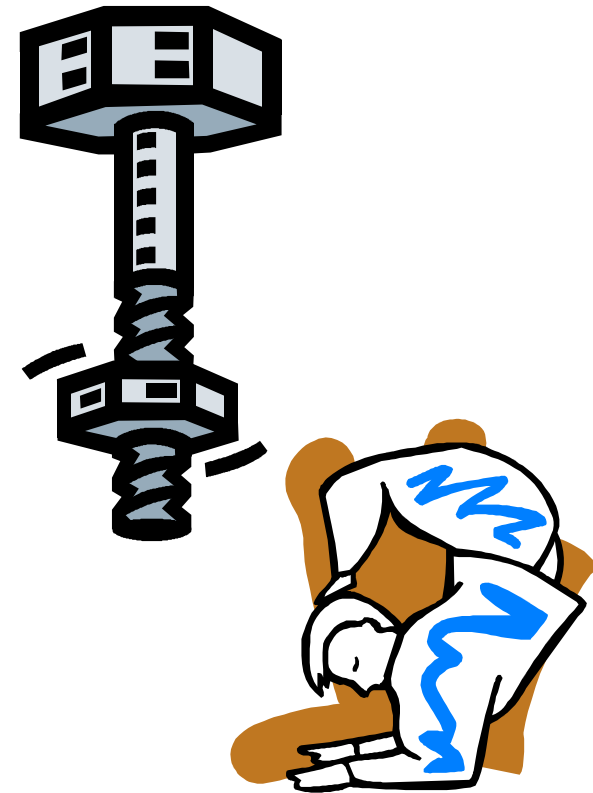
NWK

MAC/LLC

PHY

# Уровень MAC

- q Уровень MAC (Medium Access Control ) расположен между физическим уровнем (PHY) и уровнями более высокого уровня
- q Стандартный MAC IEEE 802.15.4
- q Microchip MAC на базе IEEE 802.15.4
- q Стандартный Wi-Fi MAC IEEE 802.11



# Стандарт IEEE 802.15.4™

□ Небольшой радиус действия, малое потребление и невысокая скорость передачи данных

□ Типовая дальность 30м в помещении или 100м на открытом пространстве

□ ~20 мА рабочий, < 10 мкА ждущий

□ 250 кбит/с в диапазоне 2.4 ГГц

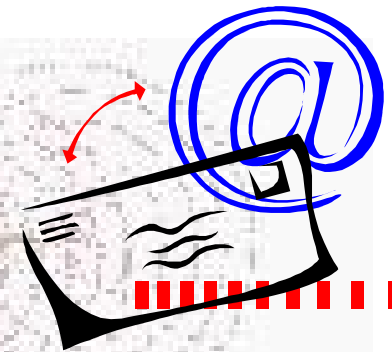


# Стандарт IEEE 802.15.4™

## q Адресация

q 8-байтный «длинный» адрес

q 2-байтный «короткий» адрес

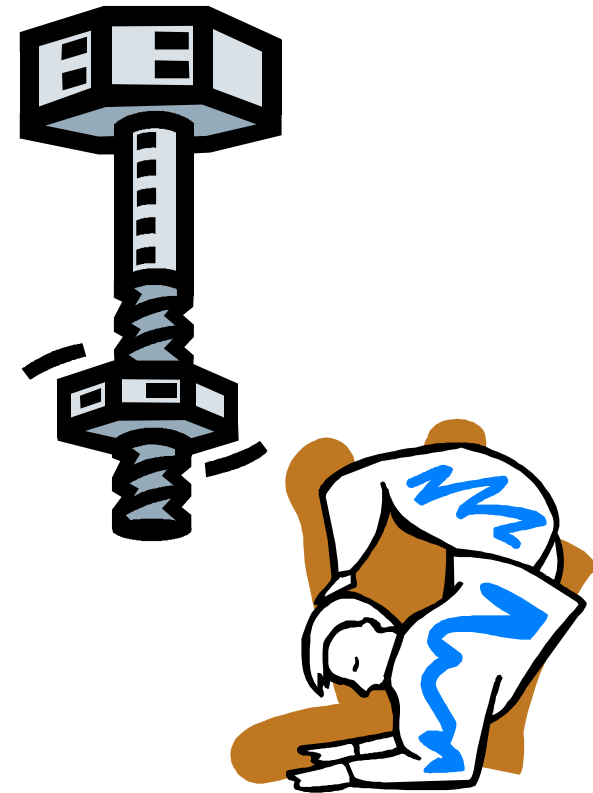


## q 7 режимов шифрования 128-bit AES



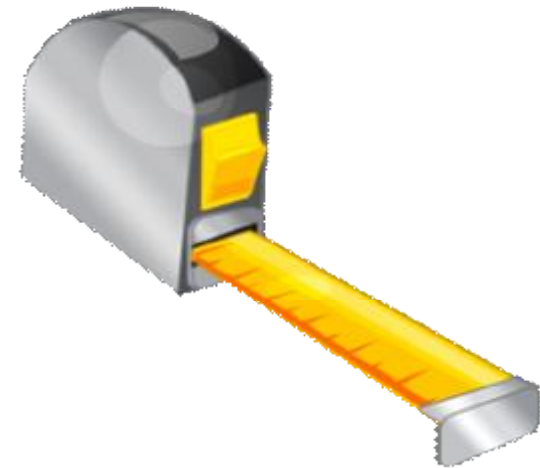
# Уровень MAC

- q Уровень MAC (Medium Access Control ) расположен между физическим уровнем (PHY) и уровнями более высокого уровня
- q Стандартный MAC IEEE 802.15.4
- q Microchip MAC на базе IEEE 802.15.4
- q Стандартный Wi-Fi MAC IEEE 802.11



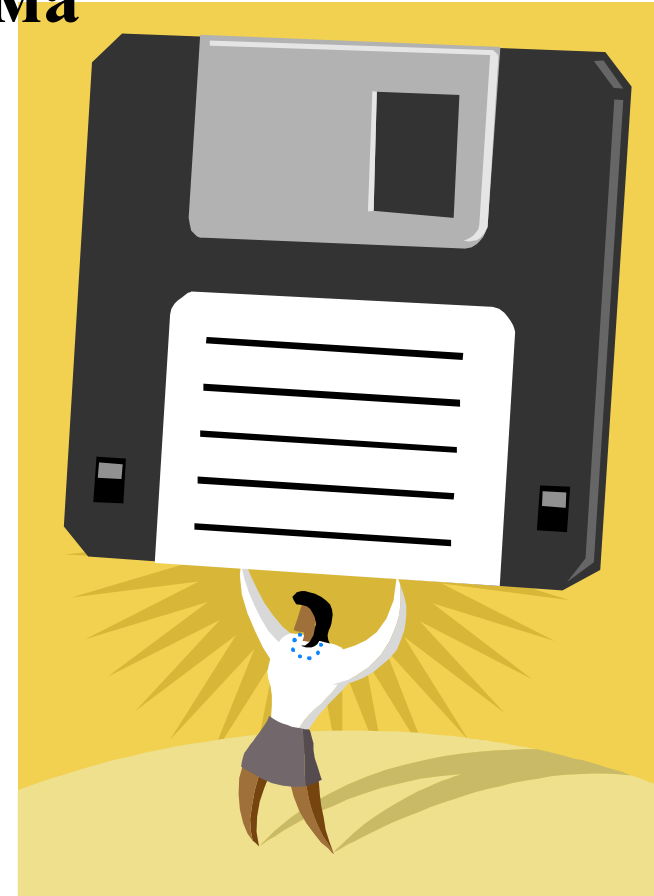
# Microchip MAC

- **Допускает работу в различных частотных диапазонах**
  - Ниже частота - больший радиус действия
- **Допускает работу с разными скоростями передачи данных**
  - Ниже скорость передачи – выше чувствительность



# Microchip MAC

- q **Используемая RF микросхема может не иметь поддержки MAC уровня**
- q **Основные функции MAC уровня реализованы программно**
  - q Индивидуальная адресация
  - q Подтверждения
  - q Повторная передача
  - q Безопасность



# Microchip MAC

## q Режим пониженного потребления

q Перевод устройства в спящий режим экономит расход энергии

q Типовая батарейка AA теоретически может обеспечивать работу устройства в течении при просыпании 1 раз в минуту.

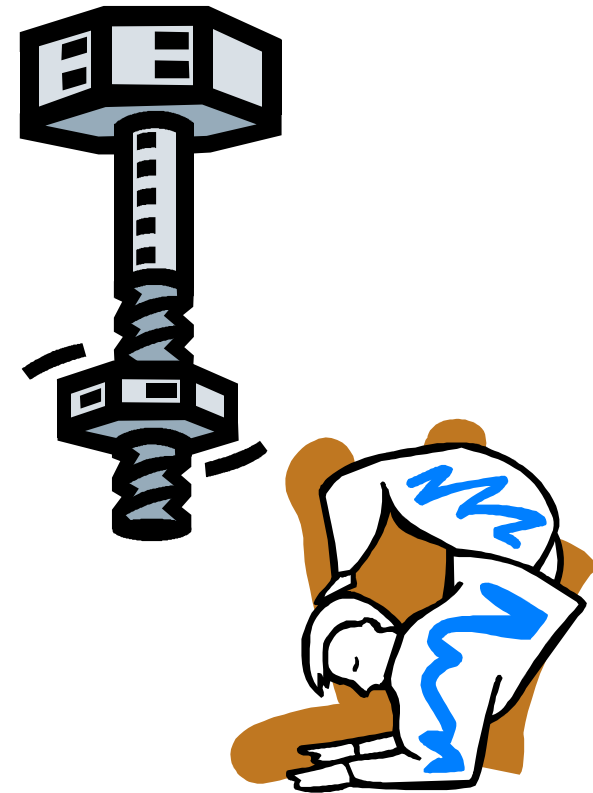
?

5 лет



# Уровень MAC

- q Уровень MAC (Medium Access Control ) расположен между физическим уровнем (PHY) и уровнями более высокого уровня
- q Стандартный MAC IEEE 802.15.4
- q Microchip MAC на базе IEEE 802.15.4
- q Стандартный Wi-Fi MAC IEEE 802.11



# Стандарт IEEE 802.11b

□ **Повышенный радиус действия, различные режимы потребления и повышенная скорость передачи данных**

□ Типовая дальность 30м в помещении или 300м на открытом пространстве

□ ~185 мА рабочий, ~250 мкА ждущий, ~ 0.1 мкА спящий

□ 1-2 Мбит/с в диапазоне 2.4 ГГц

# Стандарт IEEE 802.11b

## q Совместимость

- q Предназначен для работы в TCP/IP сетях
- q Позволяет заменить проводные решения Ethernet на беспроводные, сохранив всю инфраструктуру LAN, WAN, WWW
- q Различные аппаратные платформы имеют драйверы для поддержки WiFi





# Стандарт IEEE 802.11b

## q Режимы работы (топология)

q «Сеть» - Infrastructure (BSS)

q «Прямое соединение» Ad Hoc (IBSS)



# Удобства для разработчика

- q Поддержка стандартных протоколов
- q Легкая взаимозаменяемость протоколов
- q Легкая взаимозаменяемость трансиверов
- q Простой доступ к функциям обмена данными
- q Отсутствие необходимости опыта построения радиочастотных модулей
- q Наличие примеров применения
- q Небольшие затраты на изучение
- q Малое время и риск реализации приложения



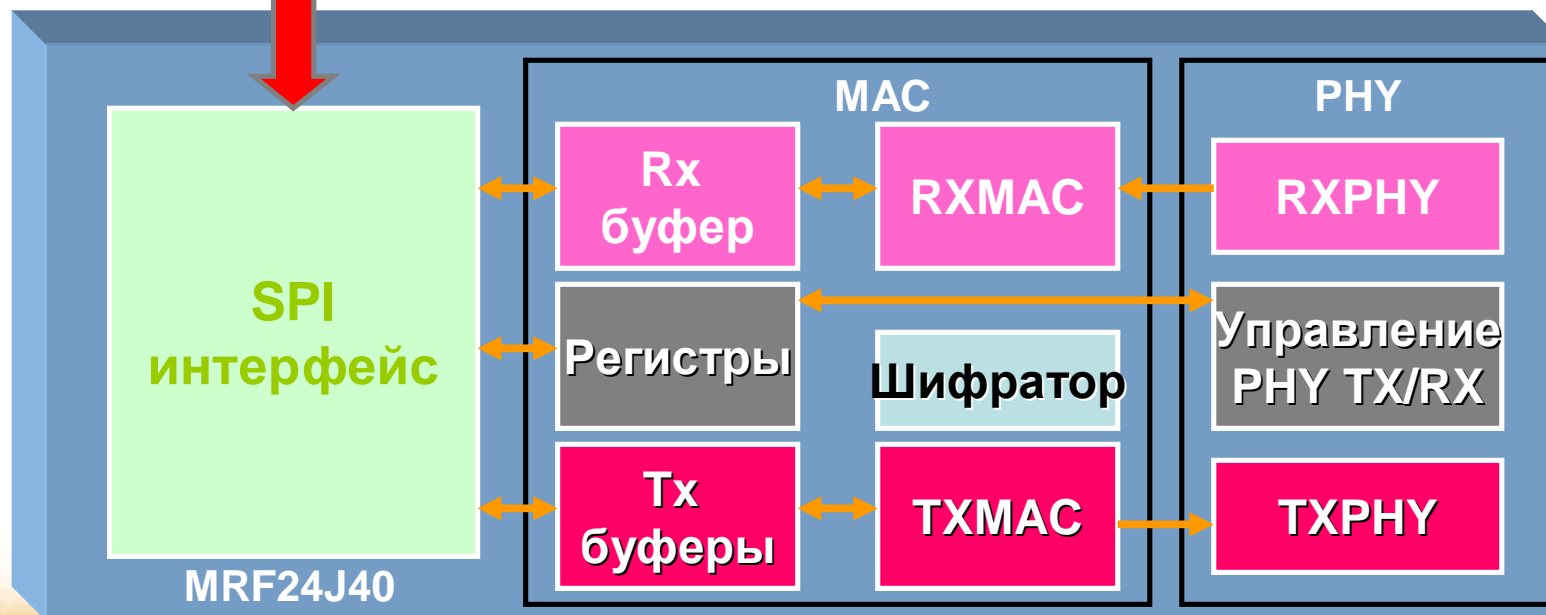
# Решения Microchip для поддержки разработки

- q **Микросхемы RF приемников/ передатчиков**
- q **Готовые, сертифицированные RF модули**
- q **Модули для отладочных плат**
- q **Стеки поддержки стандартных протоколов**
  - q MiWi P2P                      MiWi
  - q WiFi                              ZigBee
- q **Готовые демонстрационные примеры применения**
  - q MiWi P2P                      MiWi
  - q WiFi                              ZigBee

# MRF24J40



- Полностью совместим с 2.4 GHz IEEE 802.15.4™
- Поддерживает протоколы MiWi™ P2P и MiWi
- Поддерживает протокол ZigBee®
- 4-проводный SPI интерфейс
- Встроенный механизм шифрования
- Автоматическая повторная передача MAC
- 18 mA(RX)/22 mA(TX)/2 µA(Sleep)

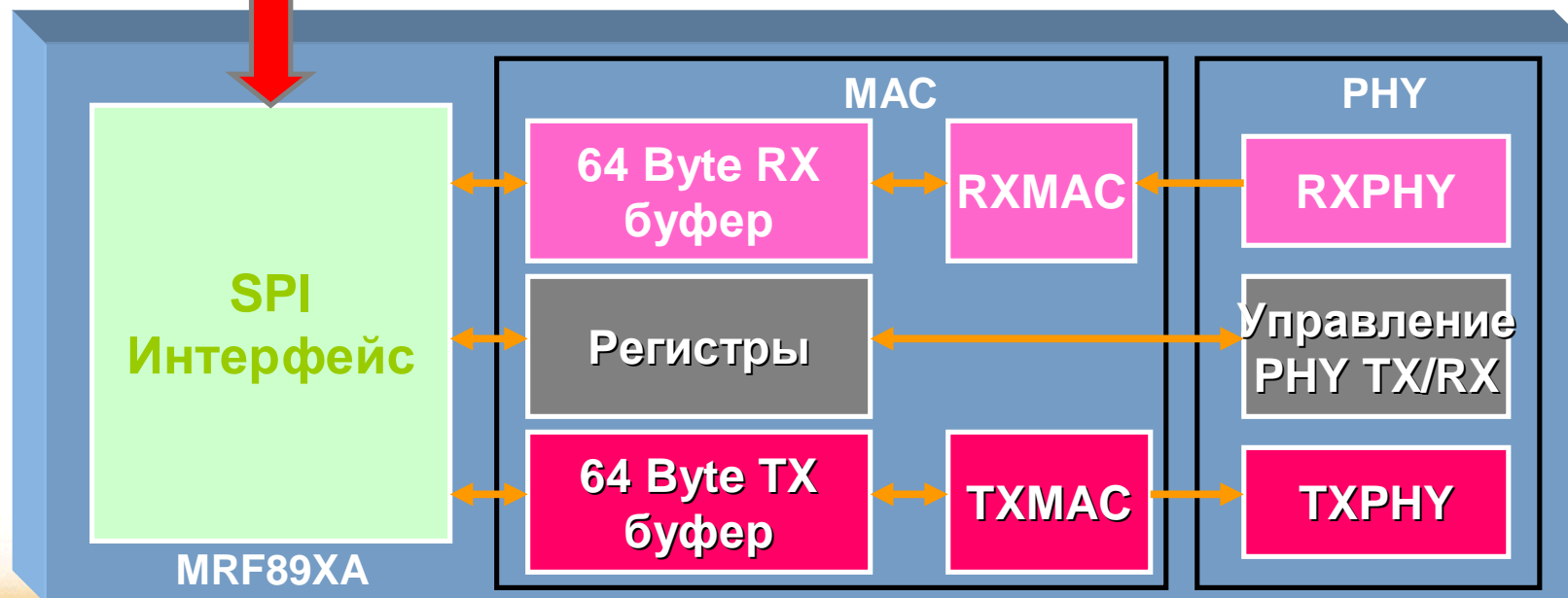




# MRF89XA Radio



- Трансивер в диапазонах 868,915 и 955МГц
- Поддерживает протоколы MiWi™ P2P и MiWi через MiApp и MiMAC
- SPI интерфейс
- Выходная мощность 10dB
- 3mA(RX) / 25mA(TX) / 100nA(Ждущий)



# Решения Microchip для поддержки разработки

- q **Микросхемы RF приемников/ передатчиков**
- q **Готовые, сертифицированные RF модули**
- q **Модули для отладочных плат**
- q **Стеки поддержки стандартных протоколов**
  - q MiWi P2P                      MiWi
  - q WiFi                              ZigBee
- q **Готовые демонстрационные примеры применения**
  - q MiWi P2P                      MiWi
  - q WiFi                              ZigBee

# RF модули Microchip

- q Законченная RF конструкция
- q Различные исполнения оптимизированных антенн
- q Различные исполнения для частотных диапазонов
- q Сертифицированы
- q SMD монтаж
- q SPI интерфейс



Это работает!





# Решения Microchip для поддержки разработки

- q **Микросхемы RF приемников/ передатчиков**
- q **Готовые, сертифицированные RF модули**
- q **Модули для отладочных плат**
- q **Стеки поддержки стандартных протоколов**
  - q MiWi P2P            MiWi
  - q WiFi                ZigBee
- q **Готовые демонстрационные примеры применения**
  - q MiWi P2P            MiWi
  - q WiFi                ZigBee

# Отладочные RF модули

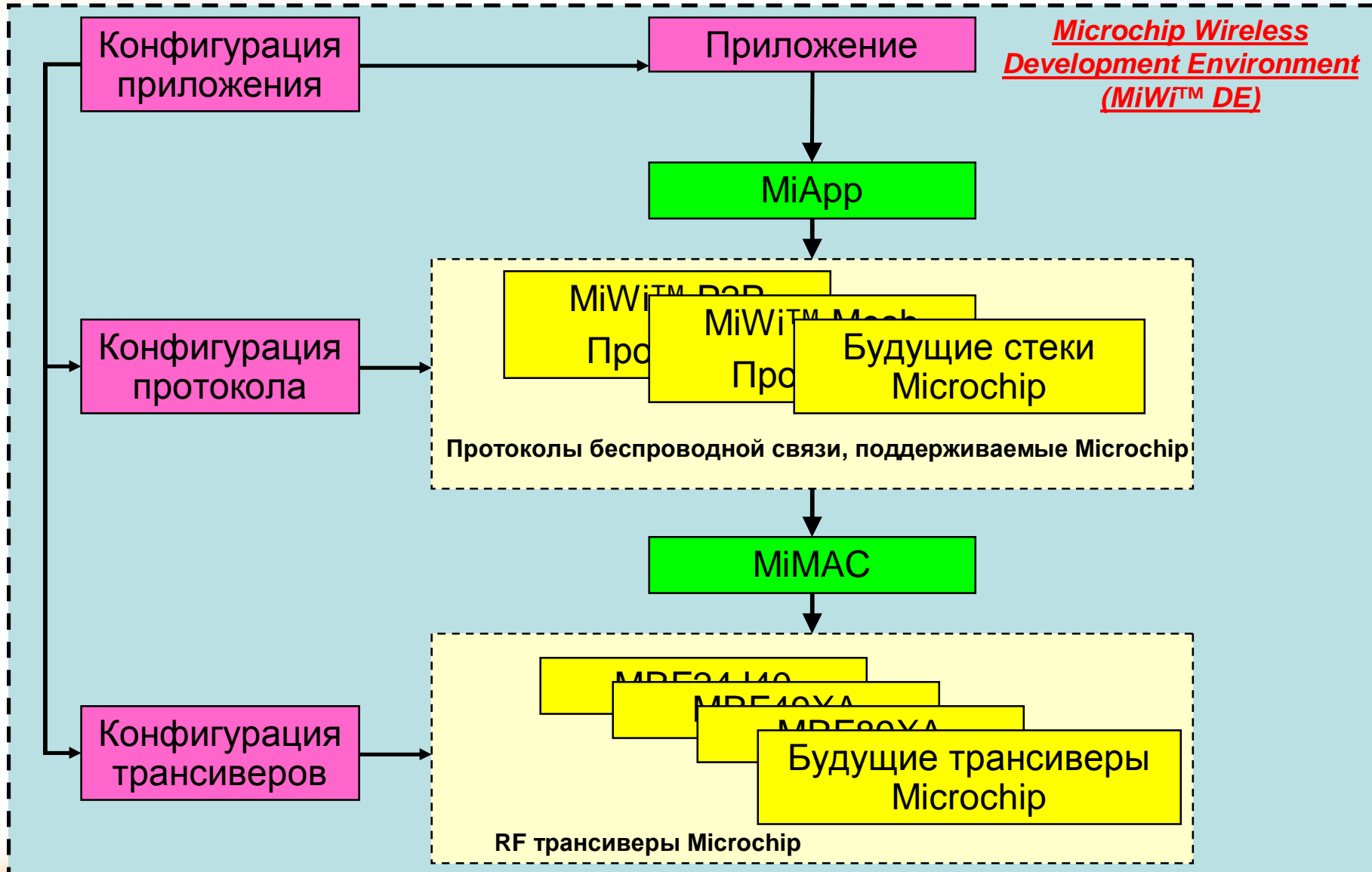
- q Исполнение PICtail / PICtail Plus
- q Подходит для отладочных плат Explorer 16 и Explorer 18



# Решения Microchip для поддержки разработки

- q **Микросхемы RF приемников/ передатчиков**
- q **Готовые, сертифицированные RF модули**
- q **Модули для отладочных плат**
- q **Стеки поддержки стандартных протоколов**
  - q **MiWi P2P      MiWi™**
  - q **WiFi**
- q **Готовые демонстрационные примеры применения**
  - q **MiWi**
  - q **WiFi**

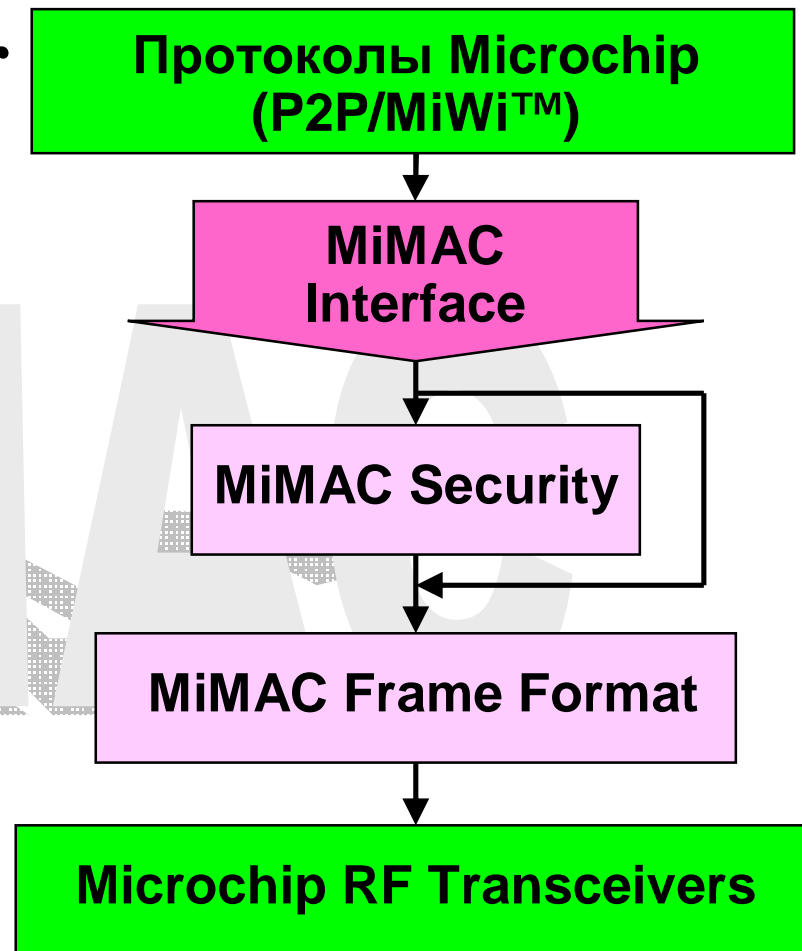
# Среда разработки MiWi™ DE



# Что такое MiMAC

q **Microchip Media Access Control (MiMAC) Layer** это MAC уровень от **Microchip**

- q Формат сообщений
- q Программный модуль безопасности/шифрования
- q Интерфейс к протоколам беспроводной связи Microchip



# Формат сообщений MiMACS

## □ Требования к формату MiMACS

- Обеспечение возможности построения сетей

- Основные характеристики сопоставимы с IEEE 802.15.4

- Обеспечение сжатых форматов кадра

- Длина MAC заголовка:

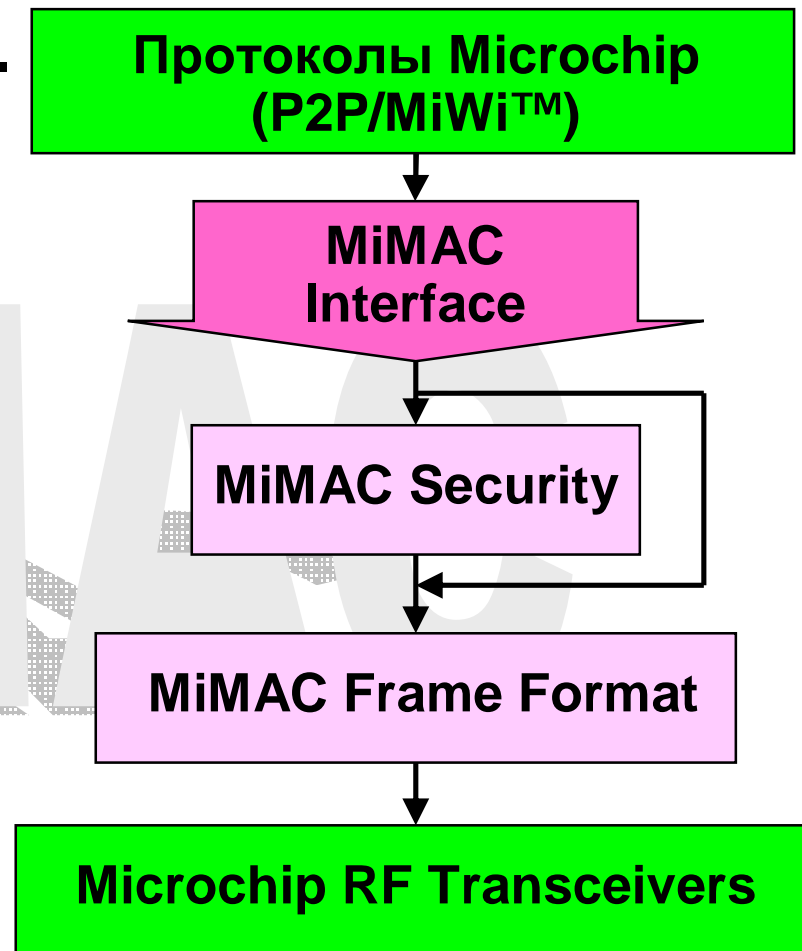
- IEEE 802.15.4: минимально **9** байт

- MiMACS: типовое **2** байт

# Что такое MiMAC

q **Microchip Media Access Control (MiMAC) Layer** - это MAC уровень от Microchip

- q Формат сообщений
- q Программный модуль безопасности/шифрования
- q Интерфейс к протоколам беспроводной связи Microchip



# Модуль безопасности MiMAC

- q **Основа модуля безопасности – алгоритм eXtended Tiny Encryption Algorithm (XTEA)**
  - q 64-битный блочный шифратор со 128-битным ключом
  - q Открытая реализация, не требует лицензирования
  - q Достаточная криптостойкость
  - q Не требует больших вычислительных ресурсов
  - q Требуемая степень криптостойкости достигается выбором нужного количества циклов шифрования

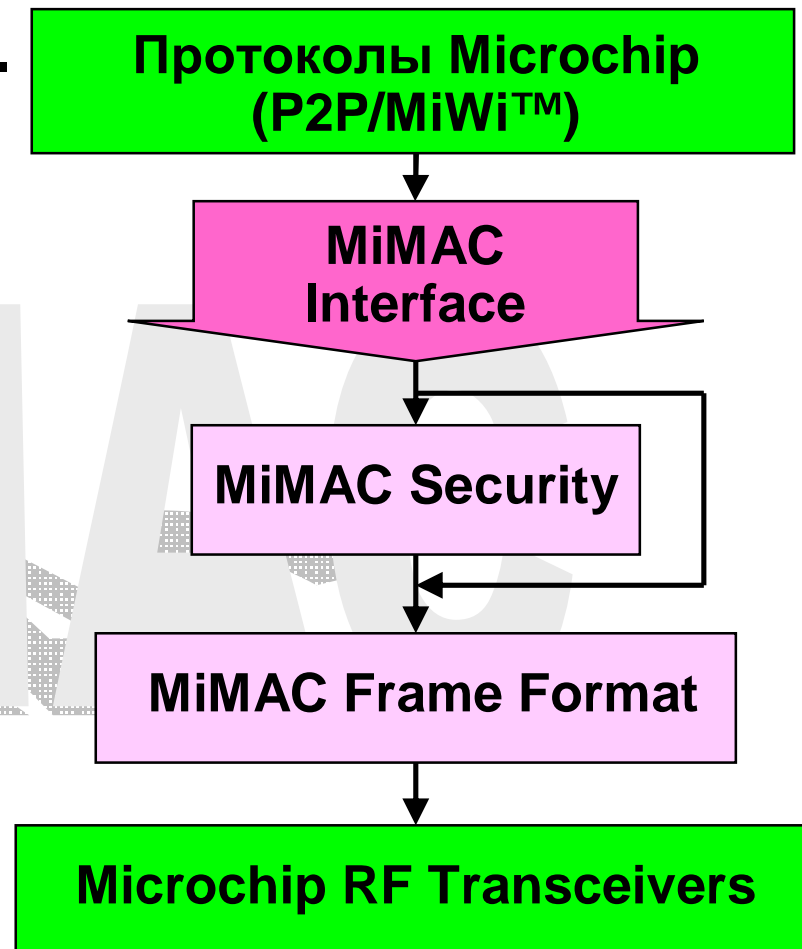




# Что такое MiMAC

q **Microchip Media Access Control (MiMAC) Layer** - это MAC уровень от Microchip

- q Формат сообщений
- q Программный модуль безопасности/шифрования
- q Интерфейс к протоколам беспроводной связи Microchip



# Интерфейс MiMAC

## q MiMAC интерфейс к протоколам беспроводной связи Microchip

- q Подходит как для трансиверов IEEE 802.15.4, так и для других трансиверов Microchip
- q Является драйвером от беспроводных протоколов Microchip к уровню MiMAC
- q Простой, но мощный

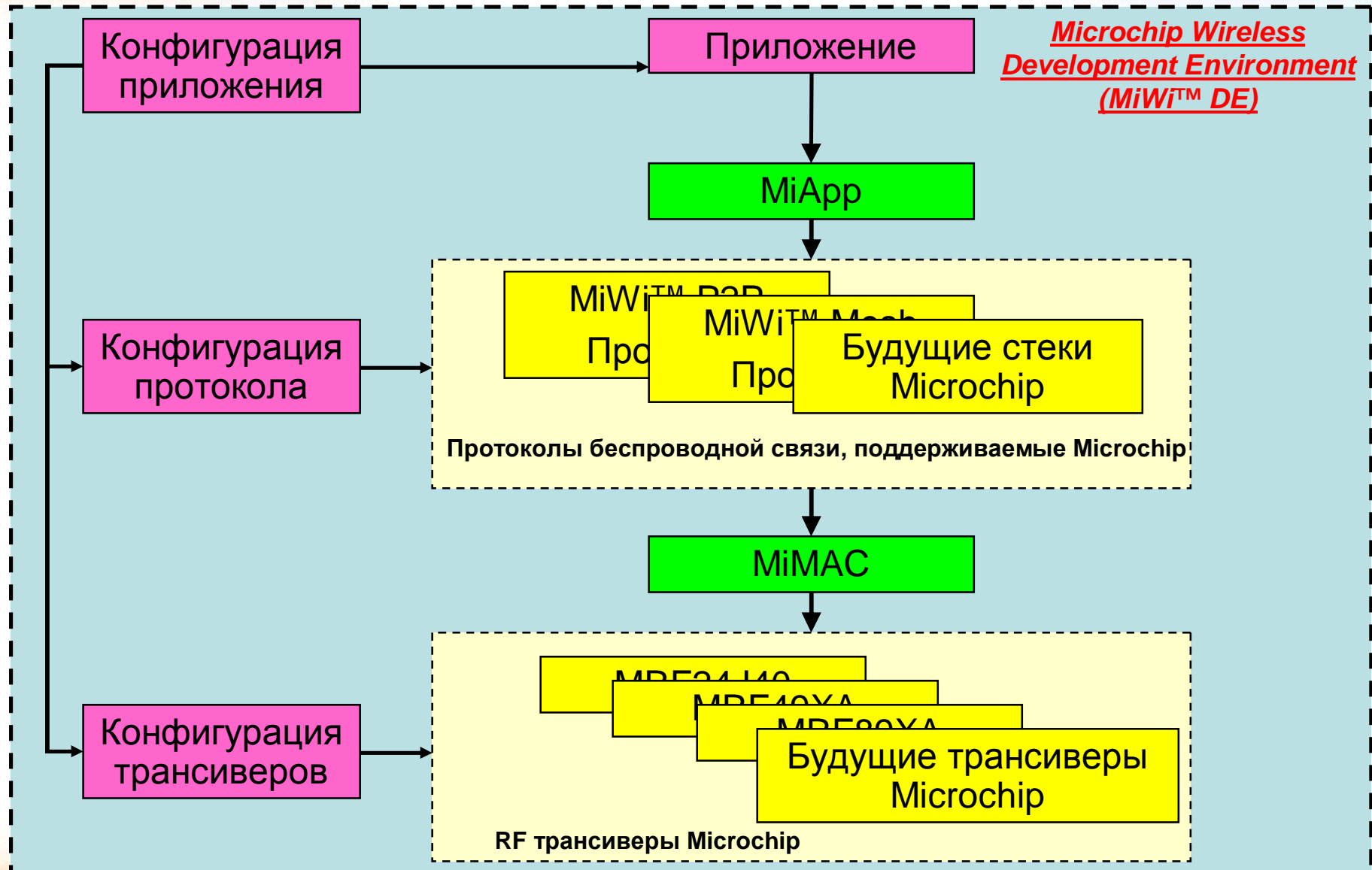


# Интерфейс MiMACS

- q Детали интерфейса MiMACS:
  - q Файл конфигурации для каждого конкретного трансивера
  - q 9 функций, вызываемых протоколами Microchip, обеспечивают:
    - q Конфигурацию
    - q TX/RX Операции
    - q Специальные функции (Спящий режим, сканирование эфира )



# Среда разработки MiWi™ DE



# MiApp API

q Четыре категории функций API

**Конфигурация**

**Установка соединения**

**Операции TX/RX**

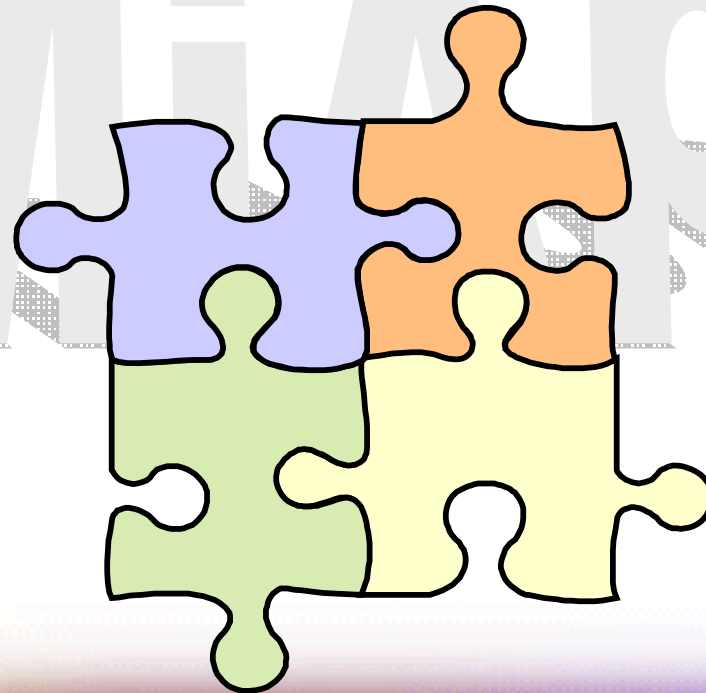
**Специальные функции**

# MiApp API

## q Конфигурация

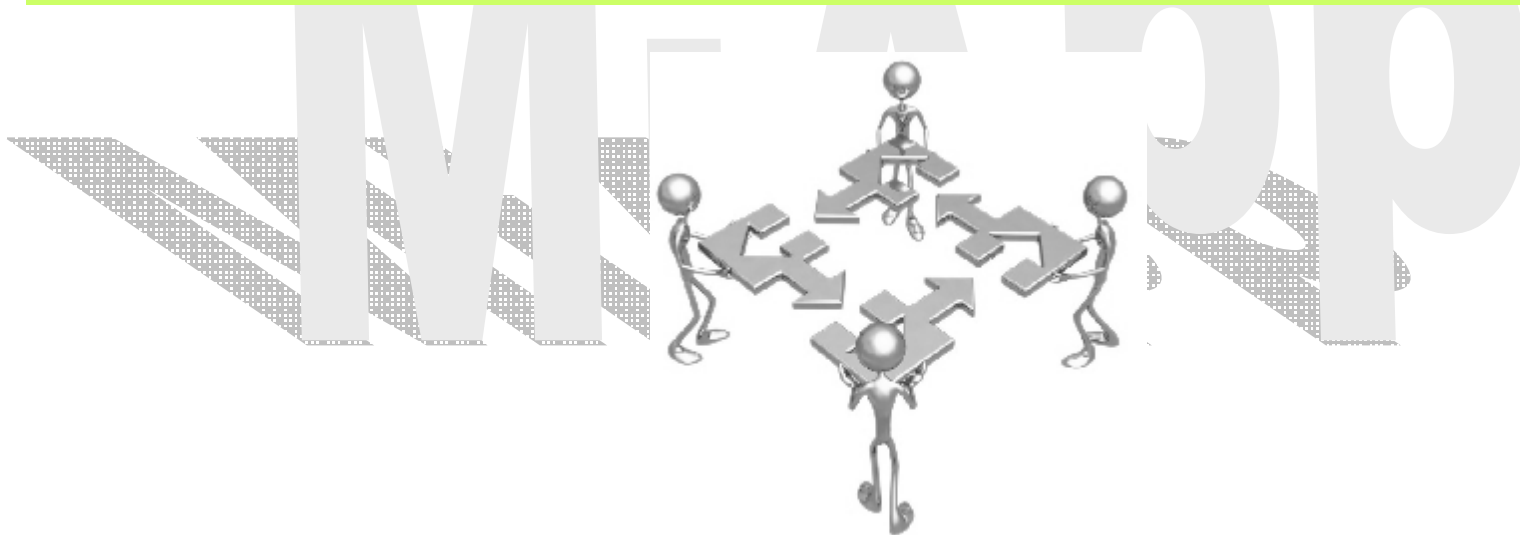
q Файл конфигурации

q Функции конфигурации



# MiApp API

- q Установка соединения
- q Режим соединения
- q Установка соединения
- q Разрыв соединения



# MiApp API

## q Операции TX/RX

### q Передача

- q Широковещательная
- q Адресная
- q Использует глобальный TX буфер

### q Прием

- q Проверить наличие сообщения
- q Игнорировать сообщения
- q Использует глобальную RX структуру

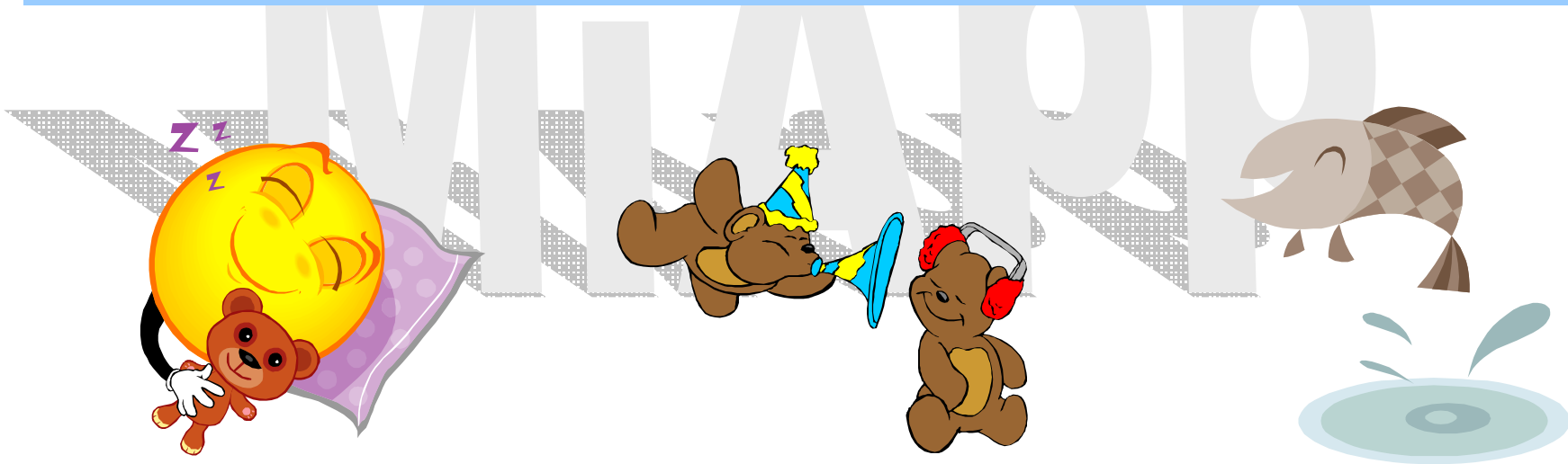




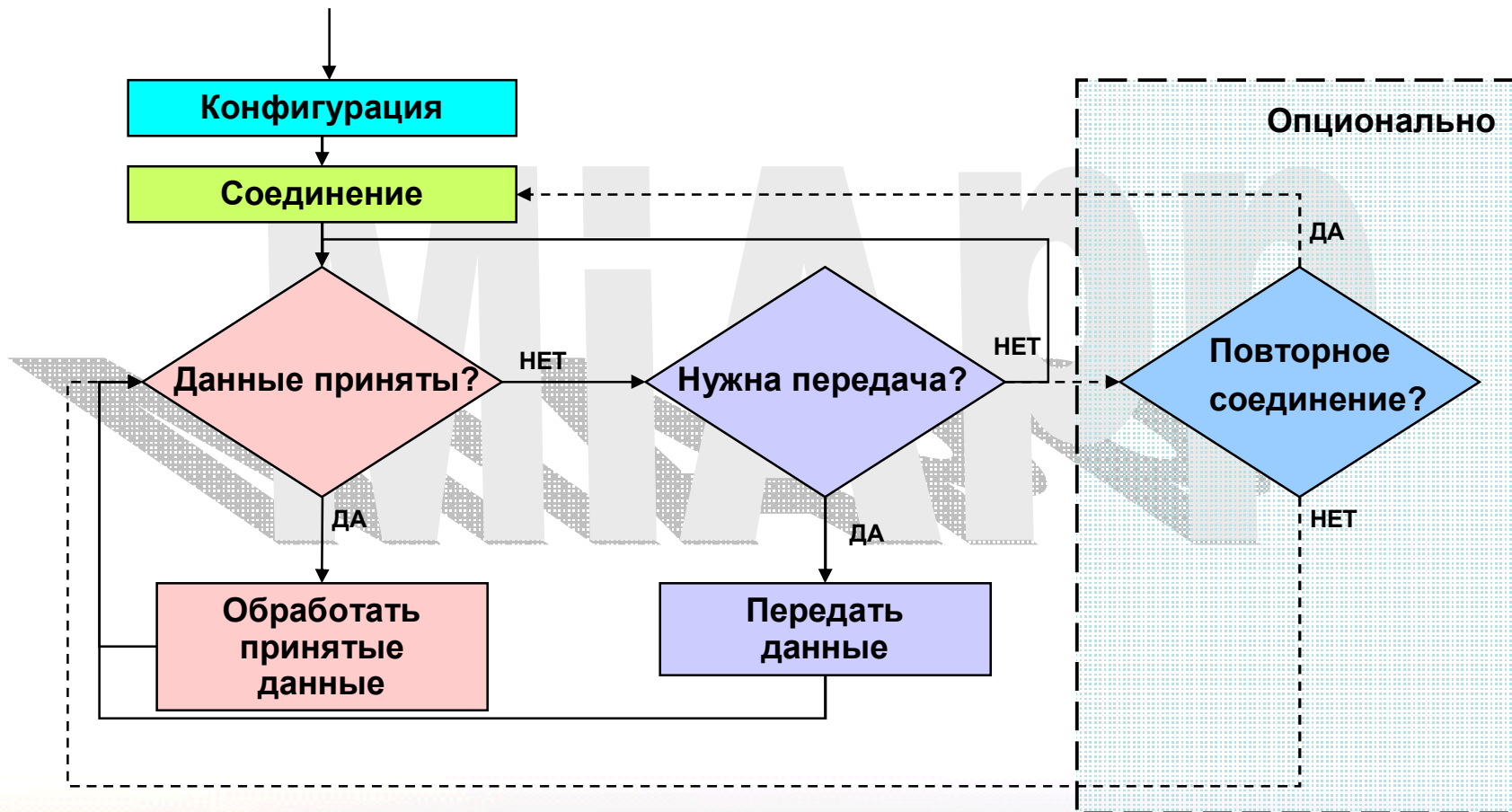
# MiApp API

## q Специальные функции

- q «Засыпание/пробуждение» трансивера
- q Распознавание шума
- q Переключение частотного канала



## q Типовое беспроводное приложение:



# Типовое приложение

```
void main(void)
```

```
{
```

1

```
    // Configuration
```

```
    MiApp_ProtocolInit(FALSE);
```

2

```
    MiApp_SetChannel(25);
```

3

```
    // Get Connected
```

```
    MiApp_ConnectionMode(ENABLE_ALL_CONN);
```

4

```
    MiApp_EstablishConnection(0xFF, CONN_MODE_DIRECT);
```

5

```
    while(1)
```

```
    {
```

6

```
        // Receive Data
```

```
        if( MiApp_MessageAvailable() )
```

```
        {
```

7

```
            LED = RxMessage.Payload[0];
```

8

```
            MiApp_DiscardMessage();
```

```
        }
```

# Типовое приложение

9

```
else if (ButtonPressed())
```

```
{
```

```
// Transmit LIGHT_ON to Peer
```

```
MiApp_FlushTx();
```

```
MiApp_WriteData(LIGHT_ON);
```

```
MiApp_UnicastConnection(0, TRUE);
```

```
}
```

```
}
```

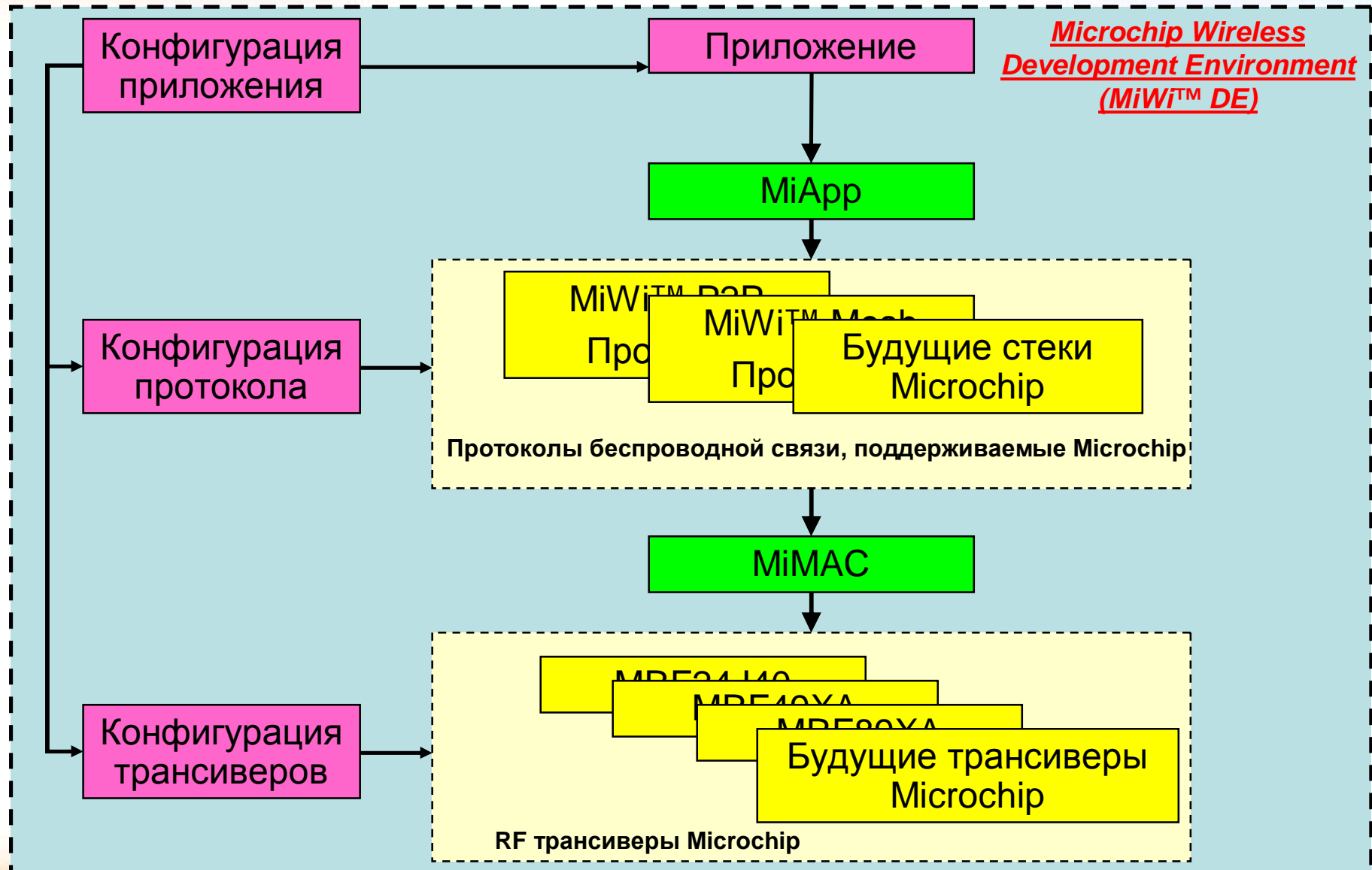
```
}
```

10

11

12

# Среда разработки MiWi™ DE



# Беспроводные протоколы Microchip

q Microchip на сегодня предлагает два беспроводных протокола:

q Протокол MiWi™ P2P

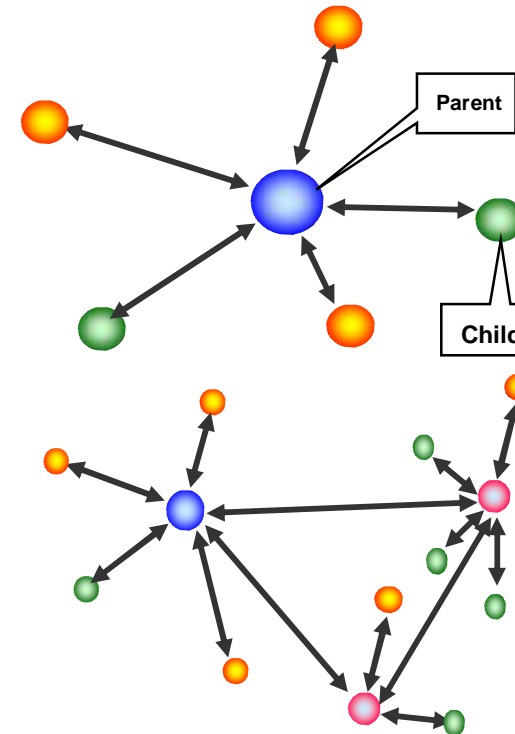
q Топология «Звезда»/P2P

q Один «прыжок», нет маршрутизации

q Протокол MiWi

q Топология «Звезда» / «Дерево»

q До 4-ч «прыжков». Маршрутизация



# MiWi™ P2P

## q Протокол MiWi™ P2P

- q Сети типа «звезда» или P2P
- q Встроенная поддержка MiMAC
- q Подходит как для трансиверов IEEE 802.15.4 , так и для других трансиверов Microchip
- q Большой набор возможностей для надежной беспроводной связи

## □ Возможности протокола MiWi P2P

### □ Гибкая конфигурация возможностей

□ Любую возможность можно включить или не включить в конфигурацию

□ Возможности могут быть включены в любом сочетании





# MiWi™ P2P

## □ Возможности протокола MiWi P2P

- Минимальный размер
- Установка соединения Hand-Shake
- Спящий режим
- Косвенная передача сообщений
- Безопасность и шифрование
- Поиск устройств ( Active Scan )
- Сканирование эфира ( Energy Scan )
- Выбор оптимального канала
- Фиксация конфигурации (Network Freezer)
- Расширенный запрос данных
- Синхронизация времени обмена

- **Протокол поддерживает все возможности протокола MiWi P2P**
- **Возможности построения Mesh сети**
- **Адресный поиск, подобный IEEE**
- **Косвенное соединение - сокет**



## □ Построение сети

- Допускает до 8 координаторов в сети и до 127 дочерних узлов на координатор
- Маршрутизация до 4-х «прыжков»



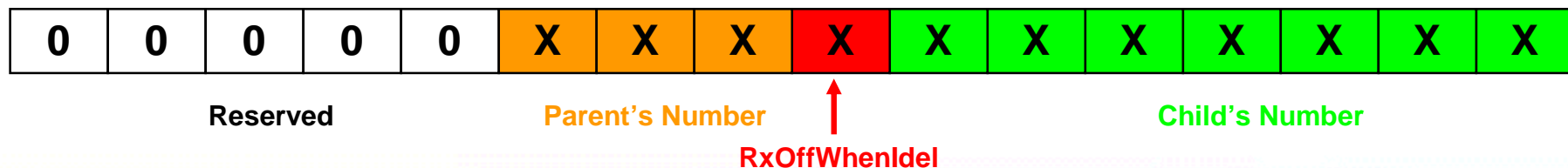
## q Универсальная адресация

q0x0000 *PAN координатор*

q0x0300 *Координатор*

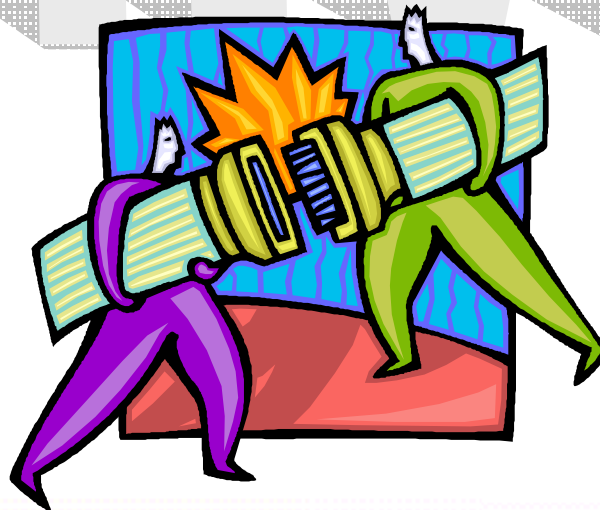
q0x0531 *Устройство с координатором 0x0500, радио всегда включено.*

q0x07A2 *Устройство с координатором 0x0700, радио выключено во время сна.*



## q Сокеты

q Сокеты – это виртуальные косвенные соединения между устройствами. Устройства могут быть вне зоны радиодоступа друг от друга.



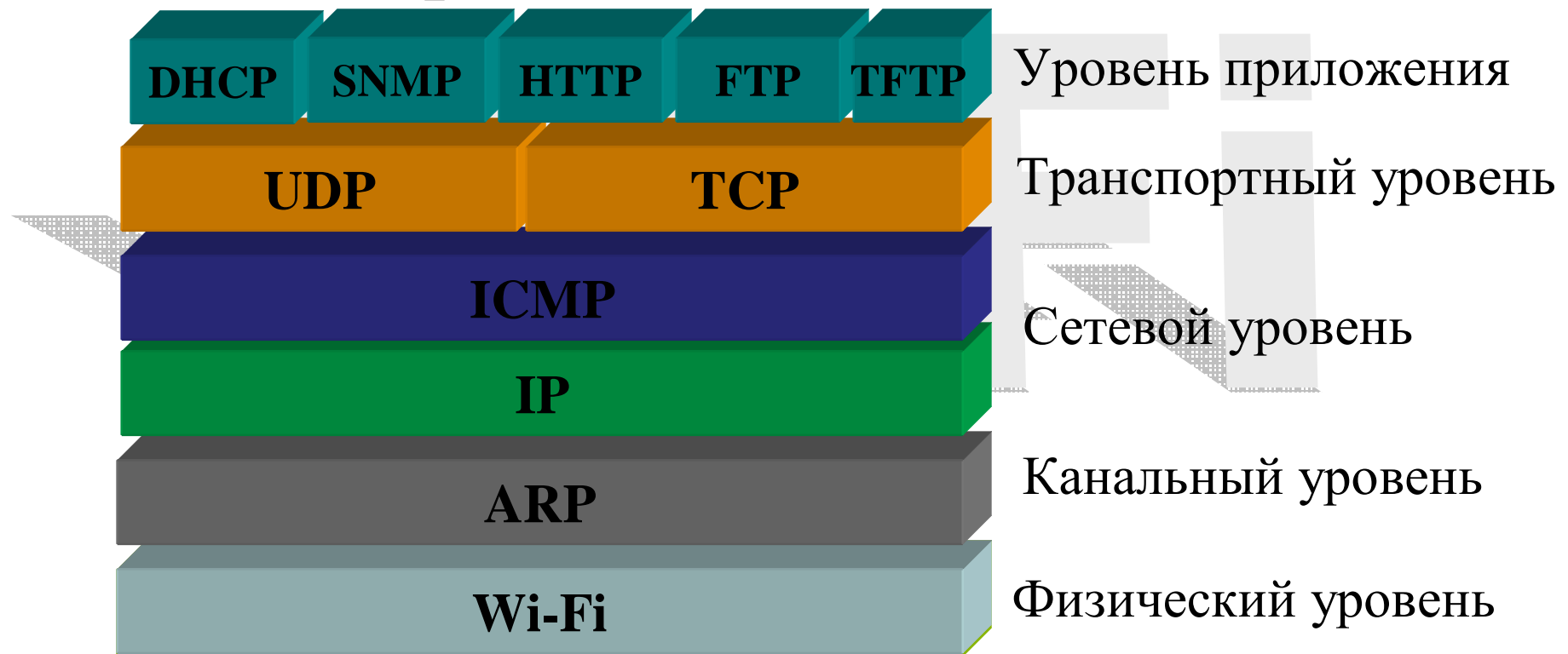
# Лабораторная работа 1

## MiWi

# Решения Microchip для поддержки разработки

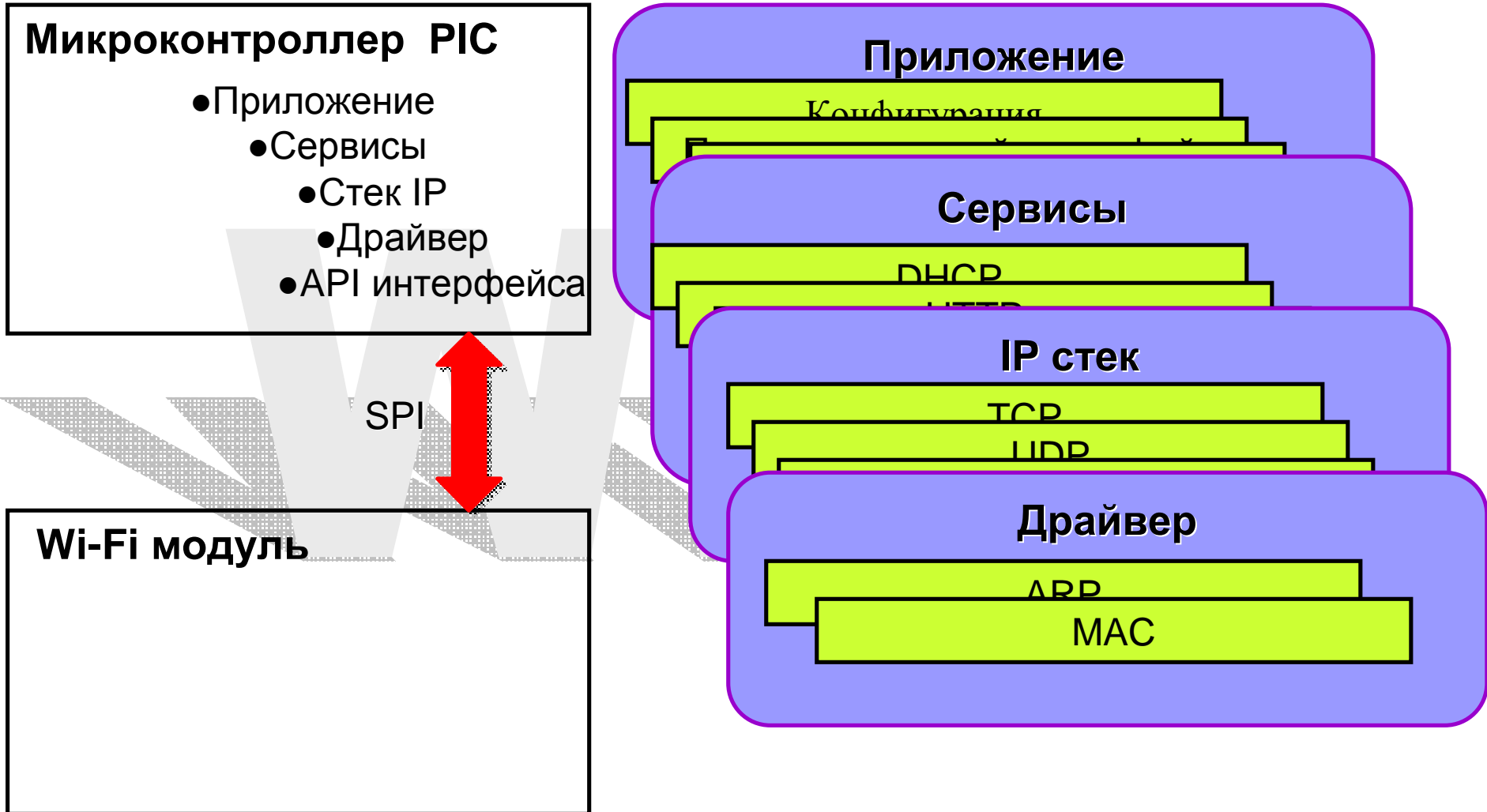
- q **Микросхемы RF приемников/ передатчиков**
- q **Готовые, сертифицированные RF модули**
- q **Модули для отладочных плат**
- q **Стеки поддержки стандартных протоколов**
  - q **MiWi P2P      MiWi™**
  - q **WiFi**
- q **Готовые демонстрационные примеры применения**
  - q **MiWi**
  - q **WiFi**

q **WiFi стек входит как составная часть в Microchip TCP/IP стек.**





# Wi-Fi®



## q Конфигурация стека TCP/IP обеспечивается:

- q Предустановленными значениями по умолчанию
- q Программой - конфигуратором
- q “С” функциями, выполняющими конфигурацию
- q Возможностями конфигурации «на лету» во время выполнения программы

# Файл конфигурации стека

## TCPIPConfig.h

```
C:\MAL\May2010\TCPIP WiFi Demo App\TCPIPConfig.h
#define STACK_USE_UART // Application demo using UART for IP address display and stack con:
//#define STACK_USE_UART2TCP_BRIDGE // UART to TCP Bridge application example
//#define STACK_USE_IP_CLEANING
#define STACK_USE_ICMP_SERVER // Ping query and response capability
//#define STACK_USE_ICMP_CLIENT // Ping transmission capability
//#define STACK_USE_HTTP_SERVER // Old HTTP server
#define STACK_USE_HTTP2_SERVER // New HTTP server with POST, Cookies, Authentication, etc.
//#define STACK_USE_SSL_SERVER // SSL server socket support (Requires SW300052)
//#define STACK_USE_SSL_CLIENT // SSL client socket support (Requires SW300052)
#define STACK_USE_AUTO_IP // Dynamic link-layer IP address automatic configuration protocol
#define STACK_USE_DHCP_CLIENT // Dynamic Host Configuration Protocol client for obtaining IP addr:
//#define STACK_USE_DHCP_SERVER // Single host DHCP server

C:\MAL\May2010\TCPIP WiFi Demo App\TCPIPConfig.h
#define MY_DEFAULT_HOST_NAME "MCHPBOARD"

#define MY_DEFAULT_MAC_BYTE1 (0x00) // Use the default of
#define MY_DEFAULT_MAC_BYTE2 (0x04) // 00-04-A3-00-00-00 if using
#define MY_DEFAULT_MAC_BYTE3 (0xA3) // an ENCX24J600 or MRF24WBOM
#define MY_DEFAULT_MAC_BYTE4 (0x00) // and wish to use the internal
#define MY_DEFAULT_MAC_BYTE5 (0x00) // factory programmed MAC
#define MY_DEFAULT_MAC_BYTE6 (0x00) // address instead.

#define MY_DEFAULT_IP_ADDR_BYTE1 (169ul)
#define MY_DEFAULT_IP_ADDR_BYTE2 (254ul)
#define MY_DEFAULT_IP_ADDR_BYTE3 (1ul)
#define MY_DEFAULT_IP_ADDR_BYTE4 (1ul)

#define MY_DEFAULT_MASK_BYTE1 (255ul)
#define MY_DEFAULT_MASK_BYTE2 (255ul)
#define MY_DEFAULT_MASK_BYTE3 (0ul)
#define MY_DEFAULT_MASK_BYTE4 (0ul)
```

Change MAC

NOTE: 00:04:A3:00:00:00

# Файл конфигурации WiFi®

## WF\_Config.h

```
C:\MAL\May2010\TCP\WiFi Demo App\WF_Config.h

/*-----*/
/* Default settings for Connection Management */
/*-----*/
#define MY_DEFAULT_SSID_NAME          "MicrochipDemoAP"

#define MY_DEFAULT_NETWORK_TYPE       WF_INFRASTRUCTURE /* WF_INFRASTRUCTURE or WF_ADHOC */

#define MY_DEFAULT_SCAN_TYPE          WF_ACTIVE_SCAN /* WF_ACTIVE_SCAN or WF_PASSIVE_SCAN */

#define MY_DEFAULT_CHANNEL_LIST       {1,6,11} /* use {} to scan all channels */

#define MY_DEFAULT_LIST_RETRY_COUNT   (3)

#define MY_DEFAULT_EVENT_NOTIFICATION_LIST (WF_NOTIFY_CONNECTION_ATTEMPT_SUCCESSFUL | \
| WF_NOTIFY_CONNECTION_ATTEMPT_FAILED | \
| WF_NOTIFY_CONNECTION_TEMPORARILY_LOST | \
| WF_NOTIFY_CONNECTION_PERMANENTLY_LOST | \
| WF_NOTIFY_CONNECTION_REESTABLISHED)

#define MY_DEFAULT_PS_POLL            WF_DISABLED /* WF_DISABLED or WF_ENABLED */

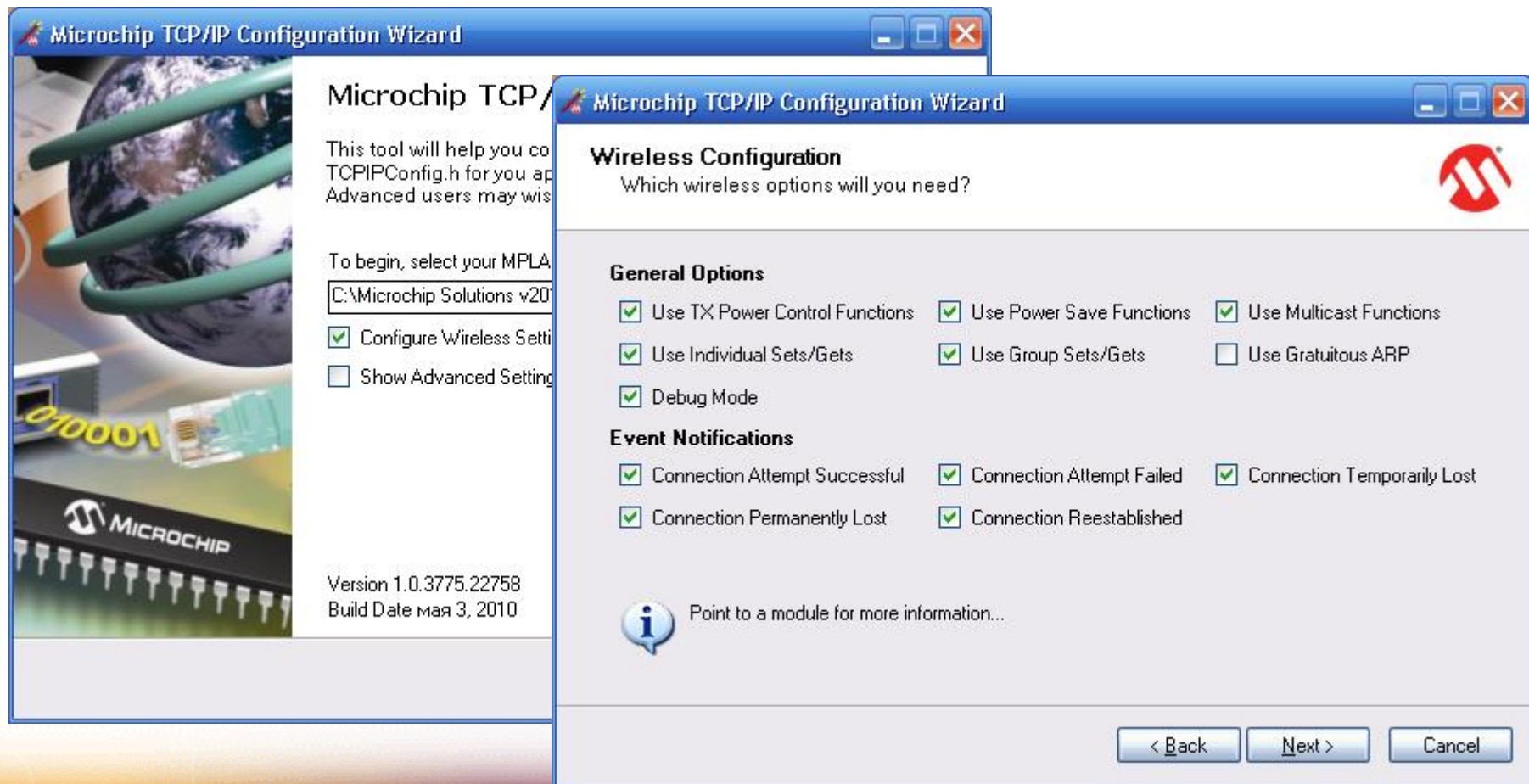
#define MY_DEFAULT_WIFI_SECURITY_MODE WF_SECURITY_OPEN

//#define USE_MRF24W_HOST_BUFFER

//#define STACK_USE_EZ_CONFIG
//#define EZ_CONFIG_SCAN
//#define EZ_CONFIG_STALL
//#define EZ_CONFIG_STORE
```

# Программа конфигуратор

## 1 Программа TCPIPConfig упрощает процесс конфигурации



# Опции конфигурации

- q **Поддержка клиента и сервера DHCP**
- q **AdHoc**
- q **Безопасность (WEP, WPA, WPA2)**
- q **Режимы энергосбережения**

# DHCP

- q **Host Control Protocol = назначение IP адреса**
- q **Динамическое (DHCP) или статическое**
- q **Динамическое требует сервер**
  - q По умолчанию статический IP адрес 169.254.1.1
  - q Если DHCP клиент включен, и ответ получен, происходит смена адреса
  - q Если DHCP клиент не включен, остается статический IP

# Безопасность

- q **Использование SSID не дает безопасности**
- q **Использует только WEP индекс 0**
- q **WEP-40 (64-bit WEP) требует 5 ASCII символов, что эквивалентно 10 Hex цифрам**
- q **WEP-104 (128-bit WEP) требует 13 ASCII символов, что эквивалентно 26 Hex цифрам**
- q **Open WEP – более безопасно, но не дает «авторизации» – вы не знаете правильный ли ключ введен. Модули только не могут понять друг друга**
- q **WPA требует ключа, вычисляемого по SSID**
- q **Используйте рекомендованные сайты для вычисления ключа (WEP/WPA)**



# Сети Adhoc

- Для конфигурации используйте EZConfig
- Удобны для построения временных сетей
- Для режима Adhoc единственный режим безопасности - WEP
- iPhone единственный смартфон, где это заложено изначально
- Убедитесь, что сеть всегда иницирует MRF24WB, или переведите все клиенты в режим 2Mbps



# Режимы энергопотребления

## 3 Режима

Активный Tx, Активный Rx

185mA, 85mA

Ожидания (Sleep)

250uA

Глубокого сна (Hibernate)

0.1uA



# Типы приложений

## Серверные

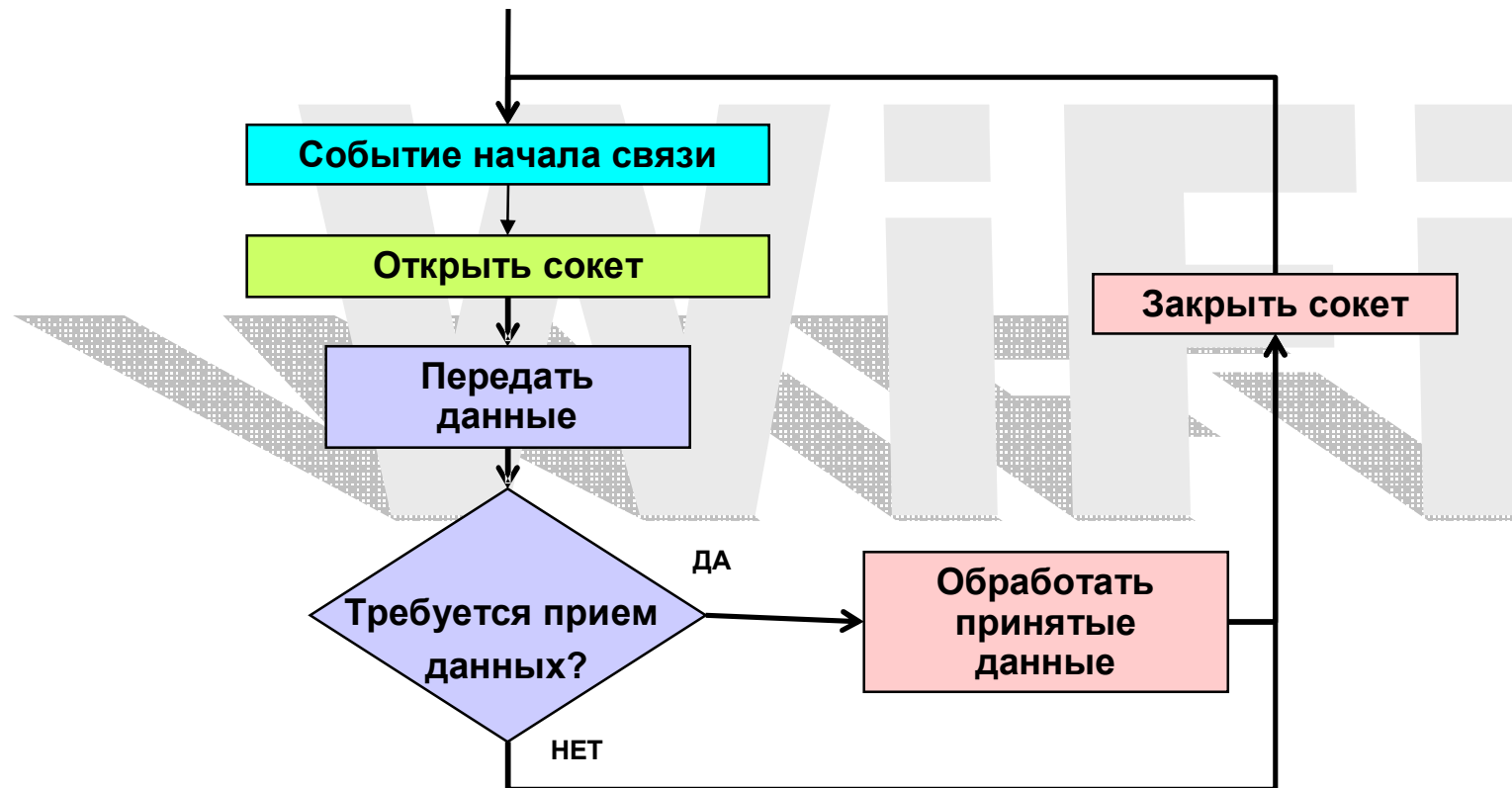
- ▣ Пассивная часть
- ▣ Обрабатывает запросы

## Клиентские

- ▣ Активная часть
- ▣ Выдает запрос в сеть

∅ Могут быть смешанные приложения, имеющие и серверную, и клиентскую часть

# Типовое клиентское приложение



# Типовое клиентское приложение

□ Приложение должно активировать машину состояний GenericTCRClient чтобы открыть соединение

# Типовое клиентское приложение

- Сокет открывается функцией `TCPOpen()`.
- Проверка установления соединения осуществляется функцией `TCPIsConnected()`

# Типовое клиентское приложение

## q Передача данных

q Функция `TCRIsPutReady()` возвращает буфер для передачи

q Функции серии `TCRPut...()` передают данные в сокет

q Формат передаваемых данных должен соответствовать формату, ожидаемому приемником информации

# Типовое клиентское приложение

- Если ожидается прием ответа, вызывается функция `TCP_IsGetReady()` чтобы определить сколько байт может быть считано из приемного буфера
- Используйте функции серии `TCP_Get...()` чтобы считать данные



# Типовое клиентское приложение

## □ Заккрытие сокета

- Вызов функции `TCPDisconnect()` закрывает соединение

- Машина состояний `GenericTCPClient` возвращается в состояние, ожидающее событие начала связи

# Лабораторная работа 2

## WiFi

# Решения Microchip для поддержки разработки

- q **Микросхемы RF приемников/ передатчиков**
- q **Готовые, сертифицированные RF модули**
- q **Модули для отладочных плат**
- q **Стеки поддержки стандартных протоколов**
  - q **MiWi P2P      MiWi™**
  - q **WiFi**
- q **Готовые демонстрационные примеры применения**
  - q **MiWi**
  - q **WiFi**

# Демонстрационные примеры MiWi

## q Простой пример

- q Автоматическая установка соединения между двумя устройствами
- q Широковещательная передача
- q Адресная передача
- q Передача зашифрованного пакета

# Демонстрационные примеры MiWi

## □ Демонстрация возможностей

- Проверка фиксации конфигурации (Network Freezer)
- Проверка поиска устройств (Active scan)
- Проверка сканирования эфира (Energy scan)
- Проверка режимов низкого потребления
- Проверка косвенной передачи сообщений
- Проверка перехода на оптимальный канал
- Проверка ресинхронизации сети после смены канала
- Проверка расширенного запроса данных
- Проверка временной синхронизации узлов

# Решения Microchip для поддержки разработки

- q **Микросхемы RF приемников/ передатчиков**
- q **Готовые, сертифицированные RF модули**
- q **Модули для отладочных плат**
- q **Стеки поддержки стандартных протоколов**
  - q **MiWi P2P      MiWi™**
  - q **WiFi**
- q **Готовые демонстрационные примеры применения**
  - q **MiWi**
  - q **WiFi**

# Демонстрационные примеры WiFi

- q Примеры использования TCP/IP стека
  - q HTTP сервер
  - q EMail
  - q TCP клиент
  - q TCP сервер
  - q Ping
  - q SNMP сервер
  - q Сети Zero Configuration

# Демонстрационные примеры WiFi

## q Примеры, специфичные для WiFi

### q Консольное демонстрационное приложение

- q Выполнение команд доступа к модулю MRF24WB0M

- q Команды управления беспроводной сетью

- q Команды управления сетью

- q Команды управления безопасностью сети

- q Пример измерения полосы пропускания и характеристик сети

- q МОЖЕТ ИСПОЛЬЗОВАТЬСЯ КАК ОТЛАДОЧНОЕ СРЕДСТВО

### q Приложение EazyConfig

- q Автоматическое назначение SSID идентификатора



# Итоги

- q Мы познакомились с решениями Microchip для беспроводных сетей MiWi
- q Мы познакомились с решениями Microchip для беспроводных сетей WiFi
- q Мы получили опыт создания приложений, использующие беспроводные решения Microchip

# Ссылки

q IEEE 802.15.4™-2003

q <http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf>

q IEEE OUI

q <https://standards.ieee.org/regauth/oui/forms/OUI-form.shtml>

q MiWi™ Протокол

q <http://www.microchip.com/MiWi>

q WiFi Протокол

q <http://www.microchip.com/WiFi>

# Спасибо!

