



# Интерфейсы связи

**USB, CAN  
Ethernet**



# USB

# Немного истории...

- | **USB была разработана при участии нескольких компаний....**
  - | Compaq
  - | Intel
  - | Microsoft
  - | NEC
  - | ...и всех, кто хотел максимально облегчить подключение периферийных устройств к компьютеру
- | **1998 – USB 1.1**
- | **2000 – USB 2.0**
- | **2003 – On-the-Go добавлено в USB 2.0 (v1.0a)**



# Виды USB-устройств

## Устройство

- Устройство предоставляет функциональность хосту
  - Например, сбор данных**

## Хаб

- Пересылает данные, контролирует питание

## Совместимое устройство

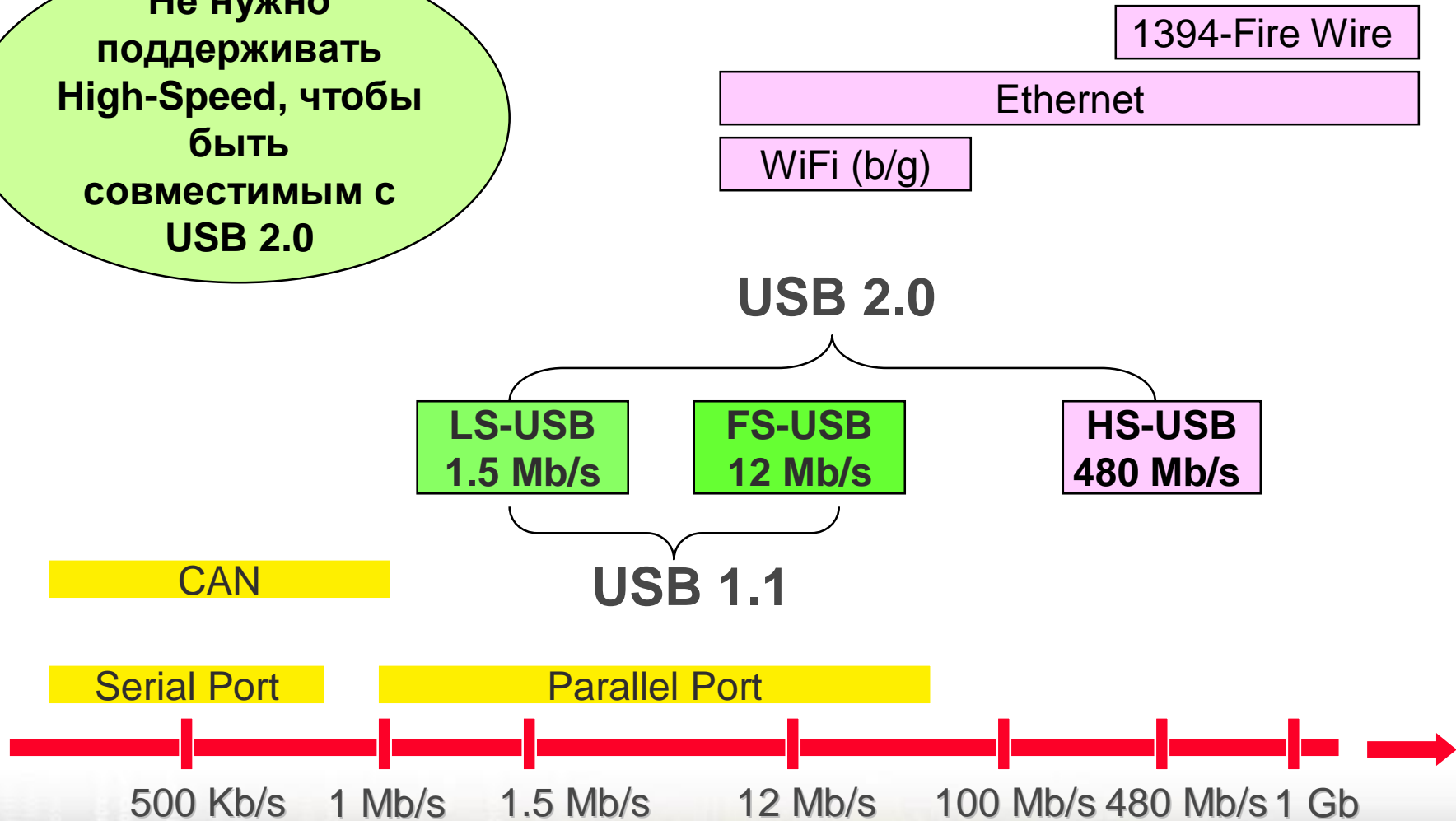
- Включает в себя хаб и 1 или несколько устройств
- Хост обращается к хабу и устройству отдельно (каждый из них имеет свой адрес)
  - Например, USB-клавиатура с однопортовым хабом**

## Композитное устройство

- Может одновременно иметь несколько активных интерфейсов
- Хост должен иметь драйвер для каждого интерфейса
  - Например, видеочамера (аудио- и видеоинтерфейсы активны)**

# Сравнение скоростей

Не нужно  
поддерживать  
High-Speed, чтобы  
быть  
СОВМЕСТИМЫМ С  
USB 2.0



# Большой миф

- | **Миф:** Низкоскоростные USB-устройства могут передавать данные со скоростью до 187.5 кбайт/с (1.5 Мбит/с)
  
- | **Факт:** Невозможно, т.к. по спецификации разрешено:
  - | Передача 8 байт каждые 10 мс
  - | Т.е. только 800 байт в секунду

# Следующий большой миф

- | **Миф:** Полноскоростные USB-устройства могут передавать данные со скоростью до 1.5 Мбайт/с (12 Мбит/с)
  
- | **Факт:** Невозможно, т.к. 1.5 Мбайт/с полная скорость ШИНЫ
  - | Она делится между устройствами
  - | Заголовки протокола
  - | Ограничения протокола
  - | Реальная скорость передачи данных одним устройством ~1.0 Мбайт/с
  - | В некоторых случаях всего 64кбайт/с

# Топология

Host (Tier 1)



USB ХОСТ

Tier 2

Keyboard

Speaker

Hub

Tier 3

Logic  
Analyzer

Hub

Tier 4

Hub

Tier 5

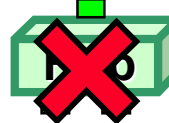
Hub

Tier 6

Hub

Tier 7

Data Logger



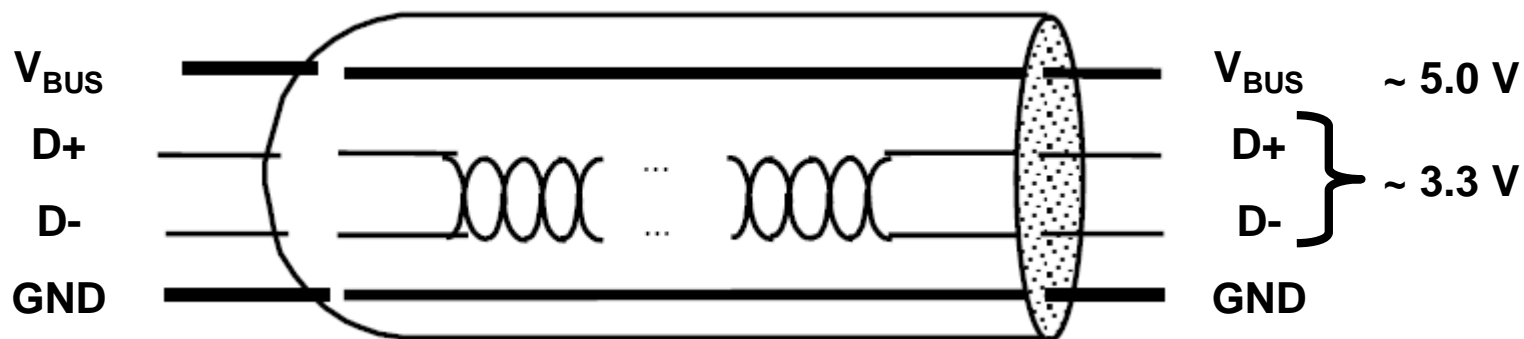
до 126 узлов

Хаб, макс.  
каскадирование = 5

PIC18 - устройство  
PIC24/PIC32 – устройство,  
Embedded host



# Физический интерфейс



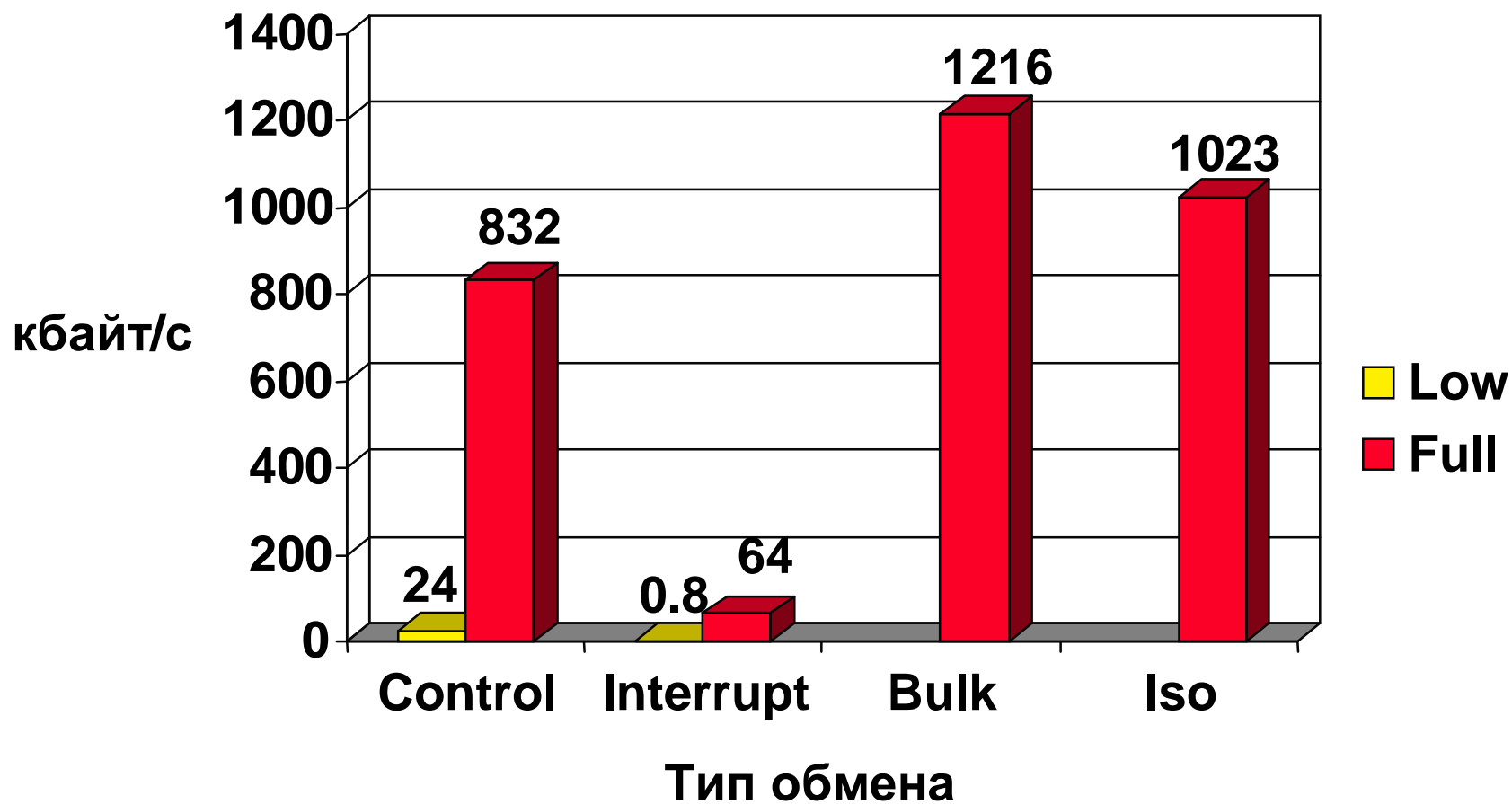
- Полудуплексная передача, кодирование данных NRZI
- Питание по шине подается на каждое устройство:
  - 4.40 - 5.25 V
  - гарантированно 100 мА
  - В некоторых случаях возможно до 500 мА

**Используйте  
внешний источник  
питания**

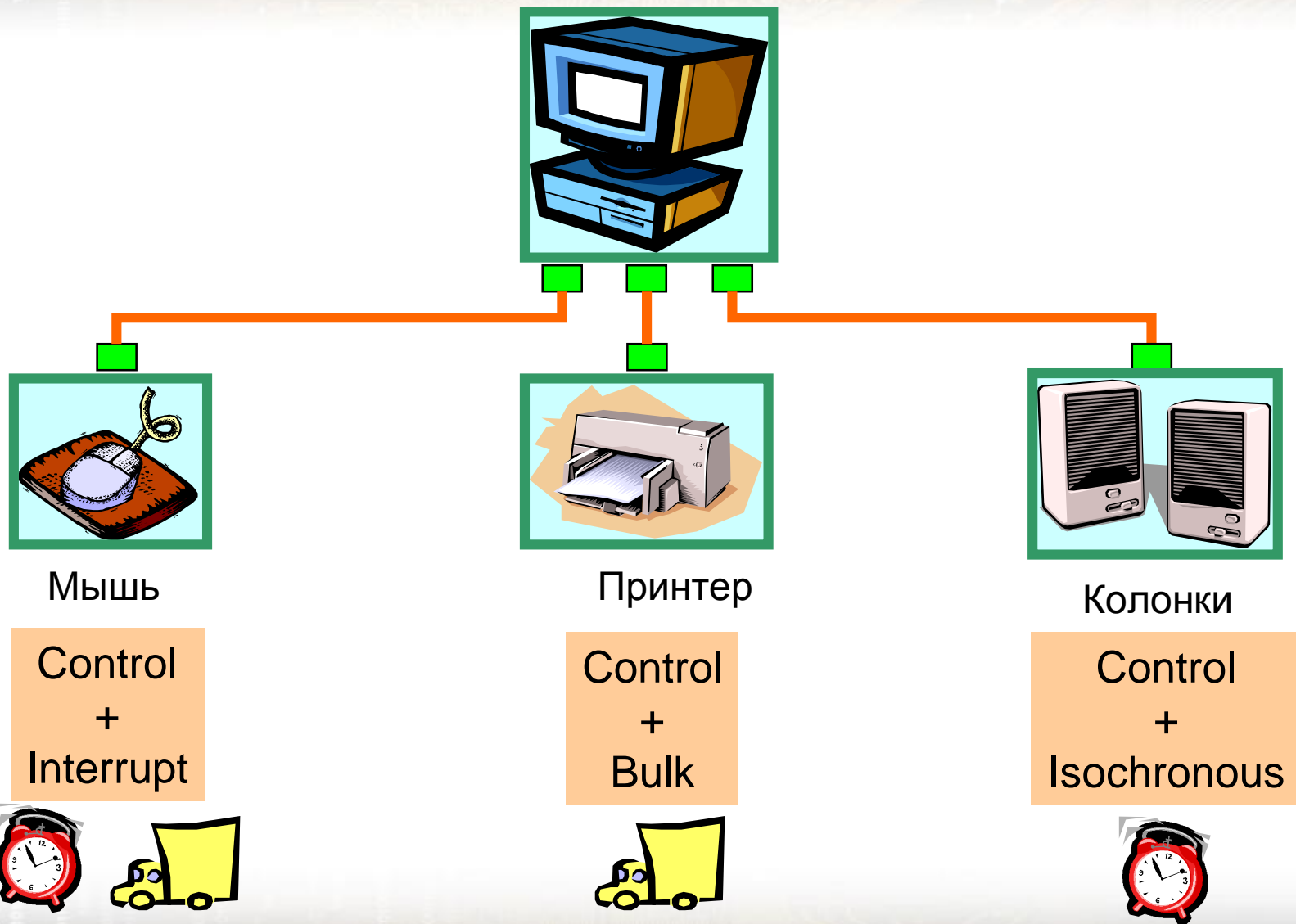
# Режимы передачи данных

<u>Тип</u>	<u>Интервал опроса</u>	<u>% использования полосы пропускания</u>	<u>Максимальный объем данных</u>	<u>Целост ность данных</u>
<b>Interrupt</b>	Фиксированный, периодический	90	64	Да
<b>Isochronous</b>	Фиксированный, периодический	90	1023	Нет
<b>Bulk</b>	Изменяемый, использует свободное время	0	1216	Да
<b>Control</b>	Изменяемый	10	832	Да

# Максимальная скорость обмена в конечной точке

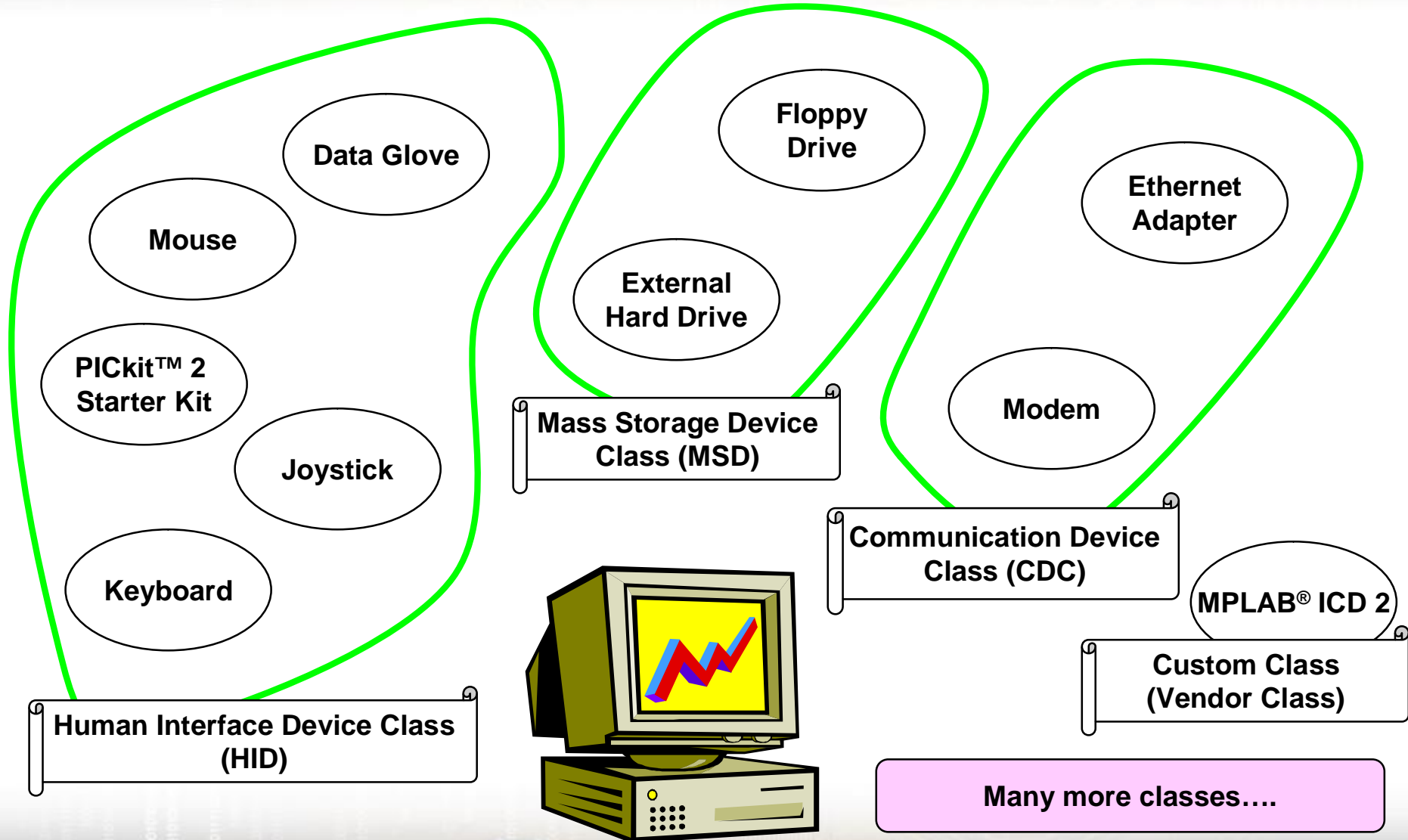


# Примеры





# Классы устройств



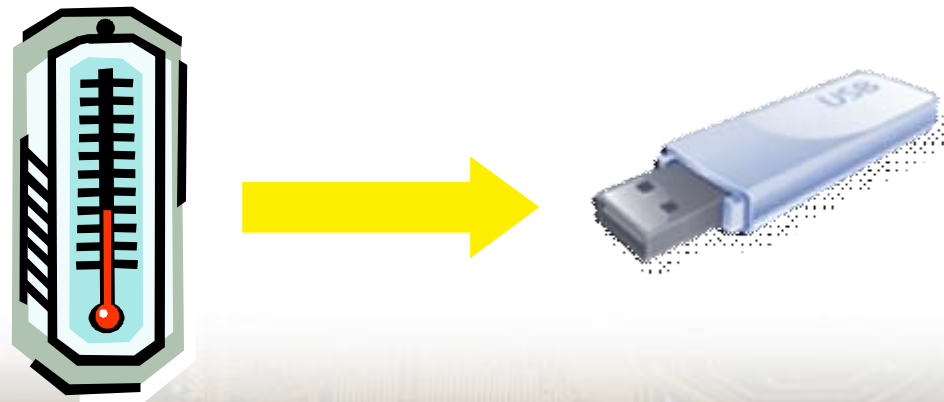
# Драйверы для Windows

Особенности	HID	CDC	mchpusb.sys (v1.1.0.0)	WinUSB
Встроенная поддержка	Да	Требуется .inf	Нет	Требуется .inf
Поддержка 64 бит	Да	Да	Да	Да
Поддержка XP	Да	Да	Да	Да
Поддержка Vista	Да	Да	Да	Да
Тип				
Control	Да	Нет	Да	Да
Interrupt	Да	Нет	Да	Да
Bulk	Нет	Да	Да	Да
Максимальная скорость	64кбайт/с	~80кбайт/с	~1.0 Мбайт/с	~1.0Мбайт/с

# USB On-The-Go

- | **Модуль USB в PIC32 работает в разных режимах:**
  - | Device mode – периферийное устройство (в этом режиме работает PIC18Fxx50)
  - | Embedded Host – работа с ограниченным набором периферийных устройств
  - | Dual-role OTG – модуль может работать и как хост, и как периферия

- | **Embedded Host**
  - | Всегда хост, никогда не периферия
  - | Всегда имеется питание 5 В
  - | Разъем типа А
- | **Пример: Система сбора данных**

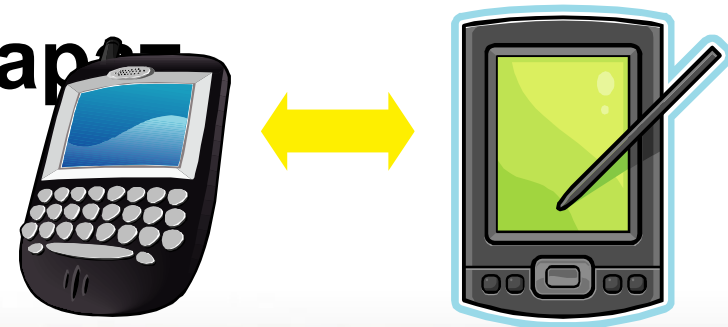




## ┆ On-The-Go (OTG)

- ┆ Мобильное устройство, простой хост
- ┆ Иногда может быть хостом, иногда периферией
- ┆ Батарейное питание
- ┆ Разъем типа микро-A/V

## ┆ Пример: КПК, фотоаппарат



# Обзор

## ┆ Dual Role

- ┆ 2 разъема – одновременно доступен для подключения ТОЛЬКО один из НИХ
- ┆ Работа и в режиме хоста, и в режиме периферии, но без динамического переключения

## ┆ Пример: Система сбора данных с возможностью обновления ПО через ПК

# Типы устройств

- ┆ **Устройство типа А**
  - ┆ Устройство, подключаемое к разъему А кабеля. Работает как ХОСТ
  
- ┆ **Устройство типа В**
  - ┆ Устройство, подключаемое к разъему В кабеля. Работает как периферия

# Разъемы

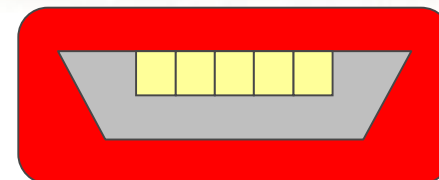
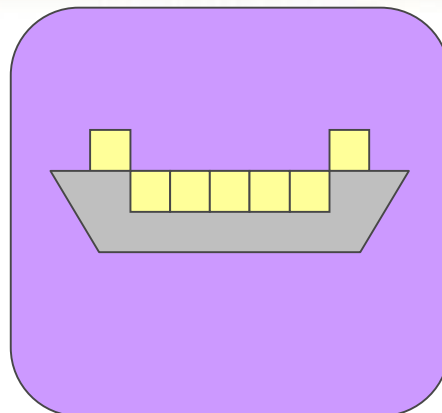
- | В классическом разъеме USB 4 контакта: VBUS, GND, D+ и D-
- | В разъеме OTG – 5 контактов, добавлен контакт ID
  - | Контакт ID предназначен для определения, какой стороной подключен кабель – со стороны хоста ID подключен к «земле»
  - | В каждом устройстве OTG контакт ID должен иметь подтяжку к «плюс» питания
  - | Внутренняя подтяжка реализована в контроллерах PIC24F и PIC32MX с USB OTG



# Разъемы

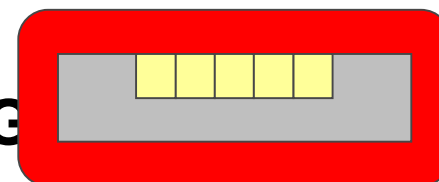
- Вилка и розетка

- микро-B

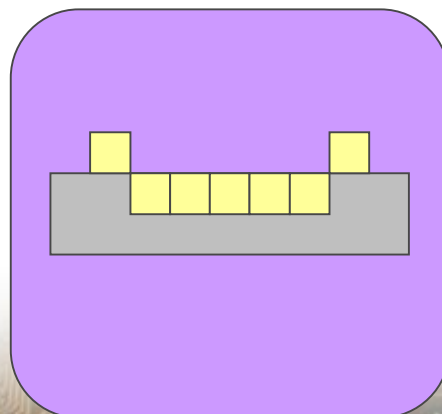


- Розетка микро-A/B

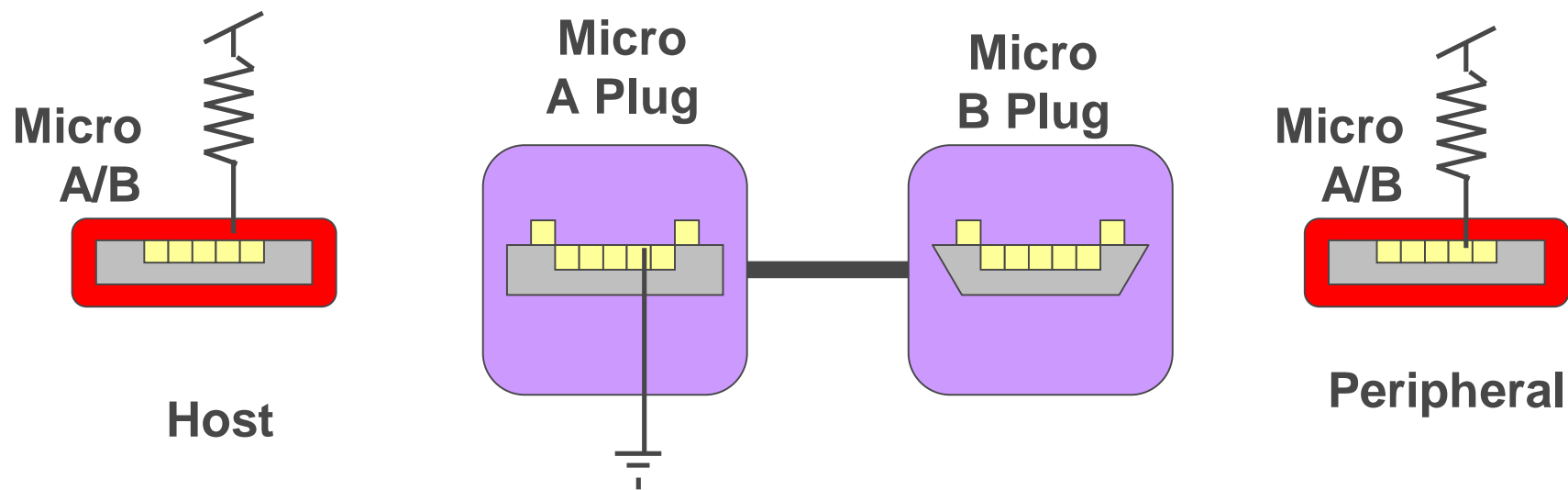
- Только для устройств с OTG



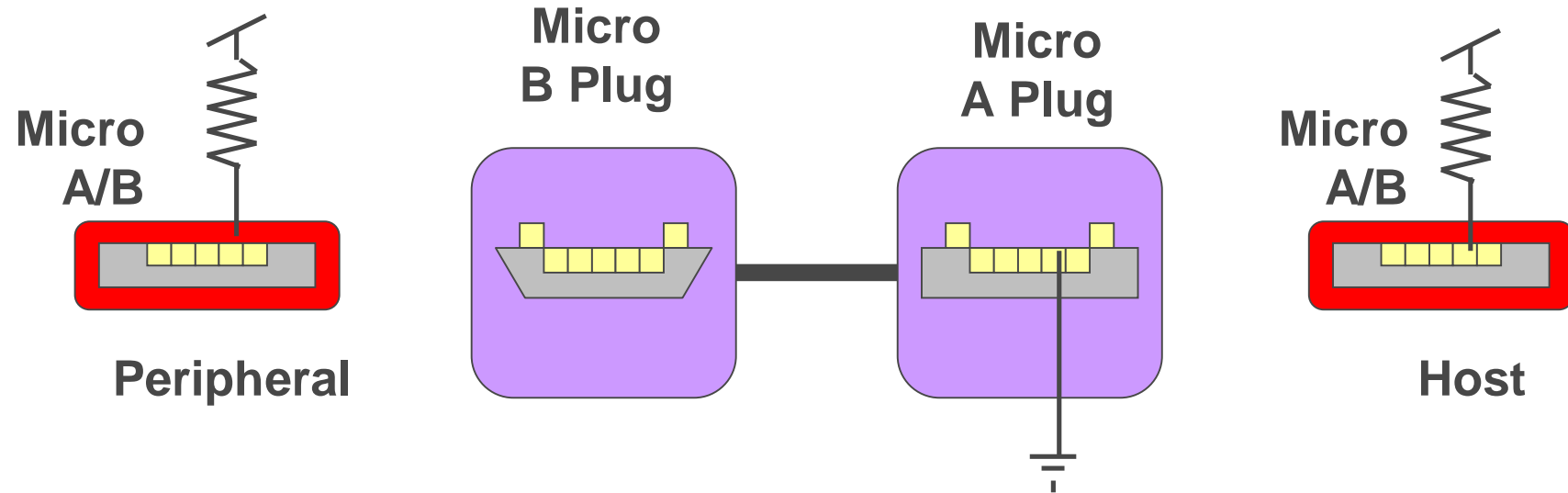
- Вилка микро-A



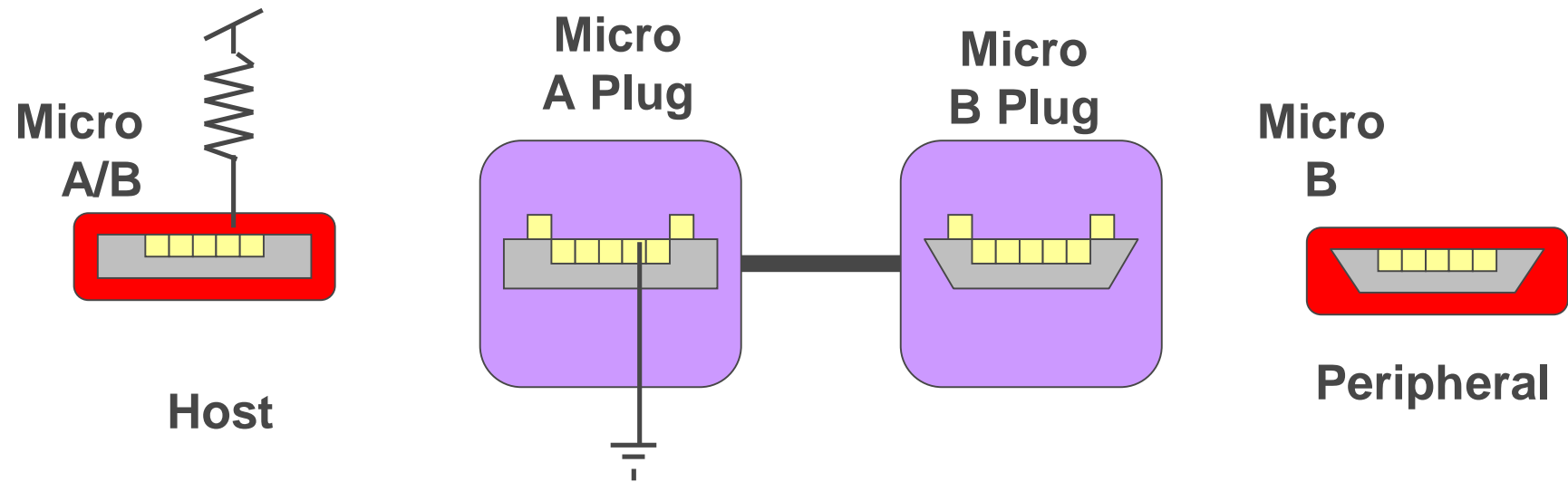
# OTG Cable Example



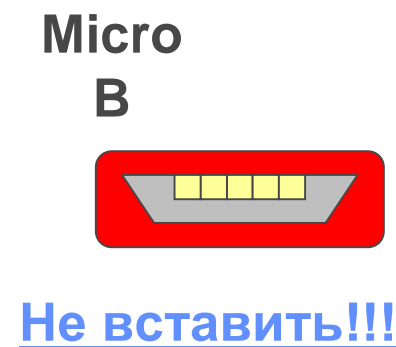
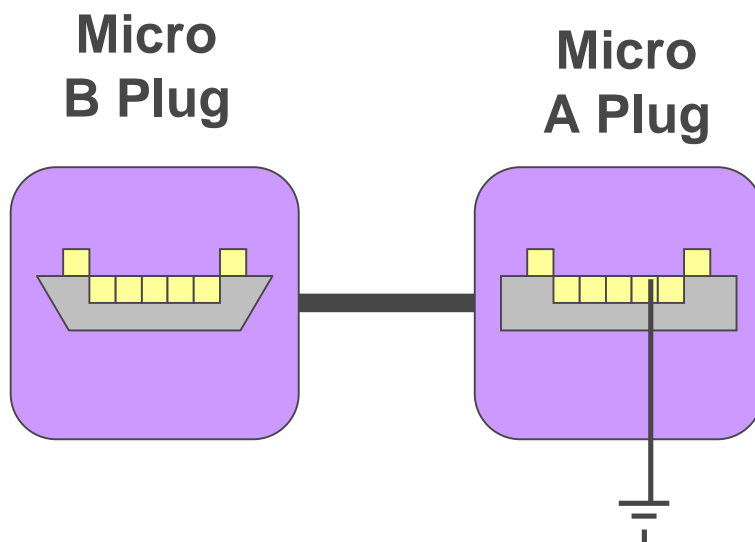
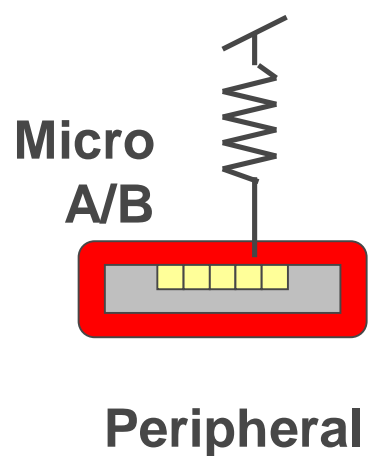
# OTG Cable Example



# OTG Cable Example



# OTG Cable Example





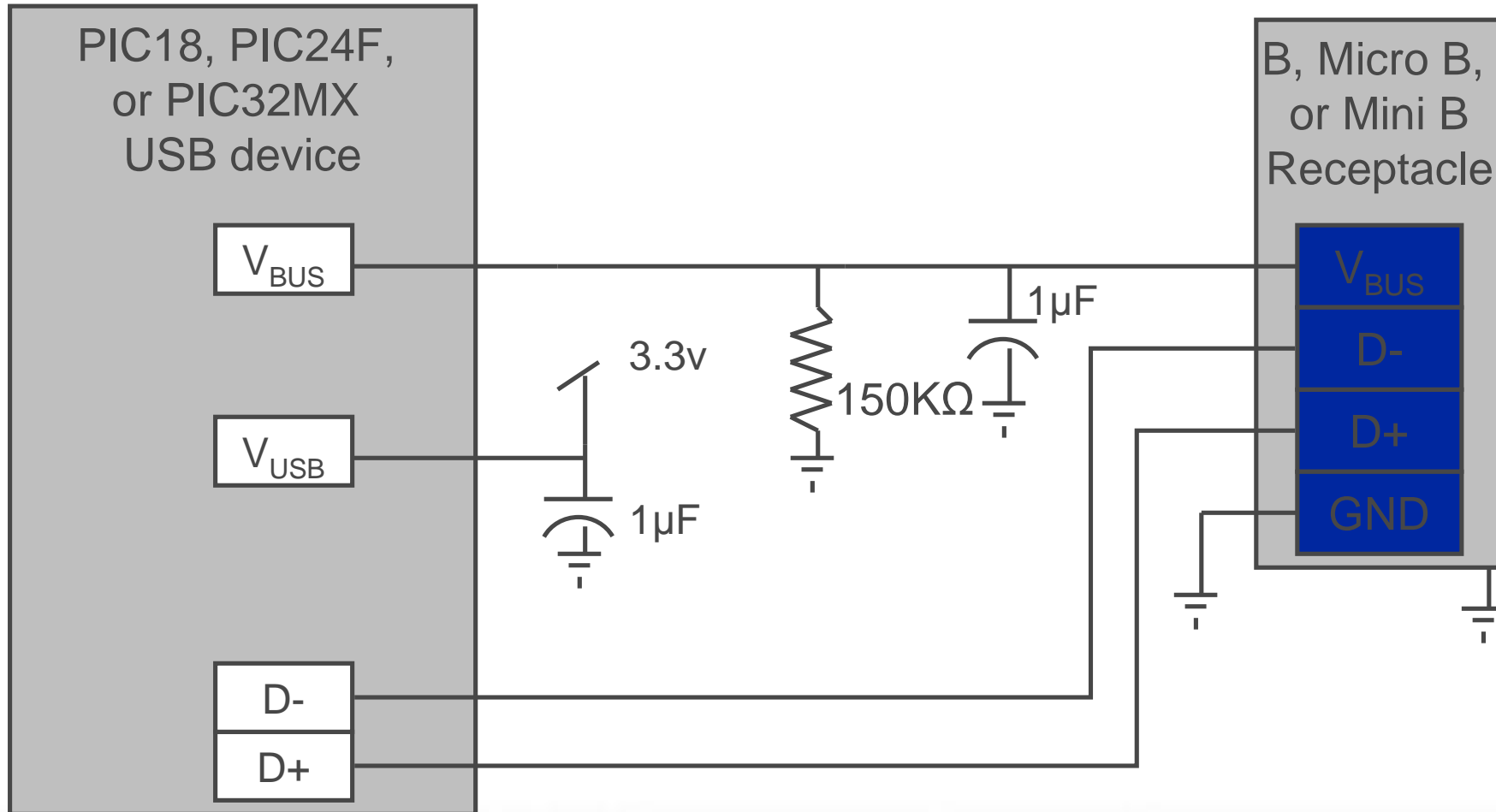
- | Для реализации OTG и Embedded Host применяются:
  - | OTG Descriptor
  - | Set Feature Requests
  - | Targeted Peripheral List (TPL)
  - | Session Request Protocol (SRP)
  - | Host Negotiation Protocol (HNP)



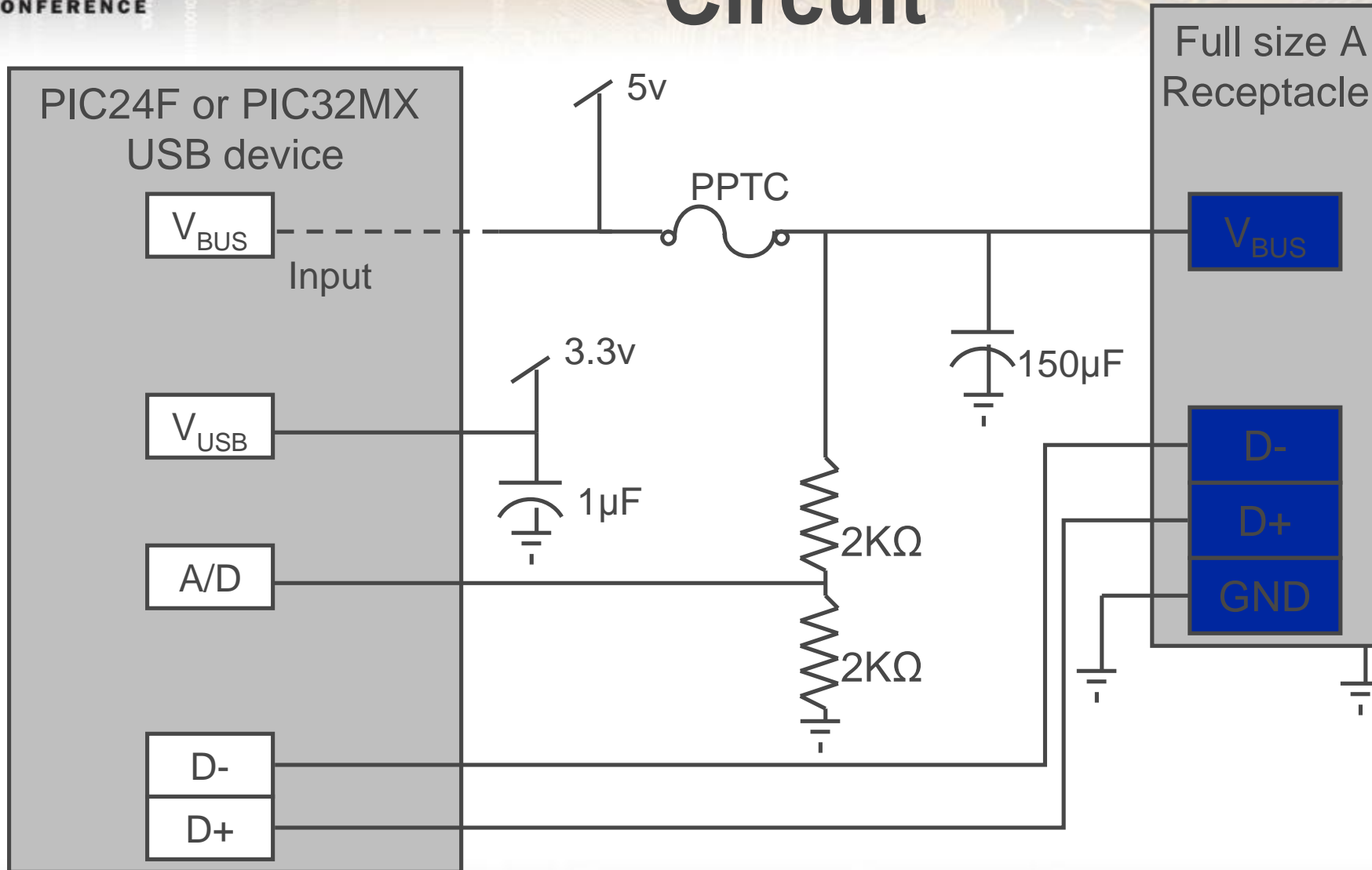
# Targeted Peripheral List (TPL)

- | **Список поддерживаемых устройств**
  - | Устройства, не содержащиеся в списке – не поддерживаются
  - | Embedded host не может поддерживать подключение устройства OTG
- | **Производитель, модель и дескриптор**

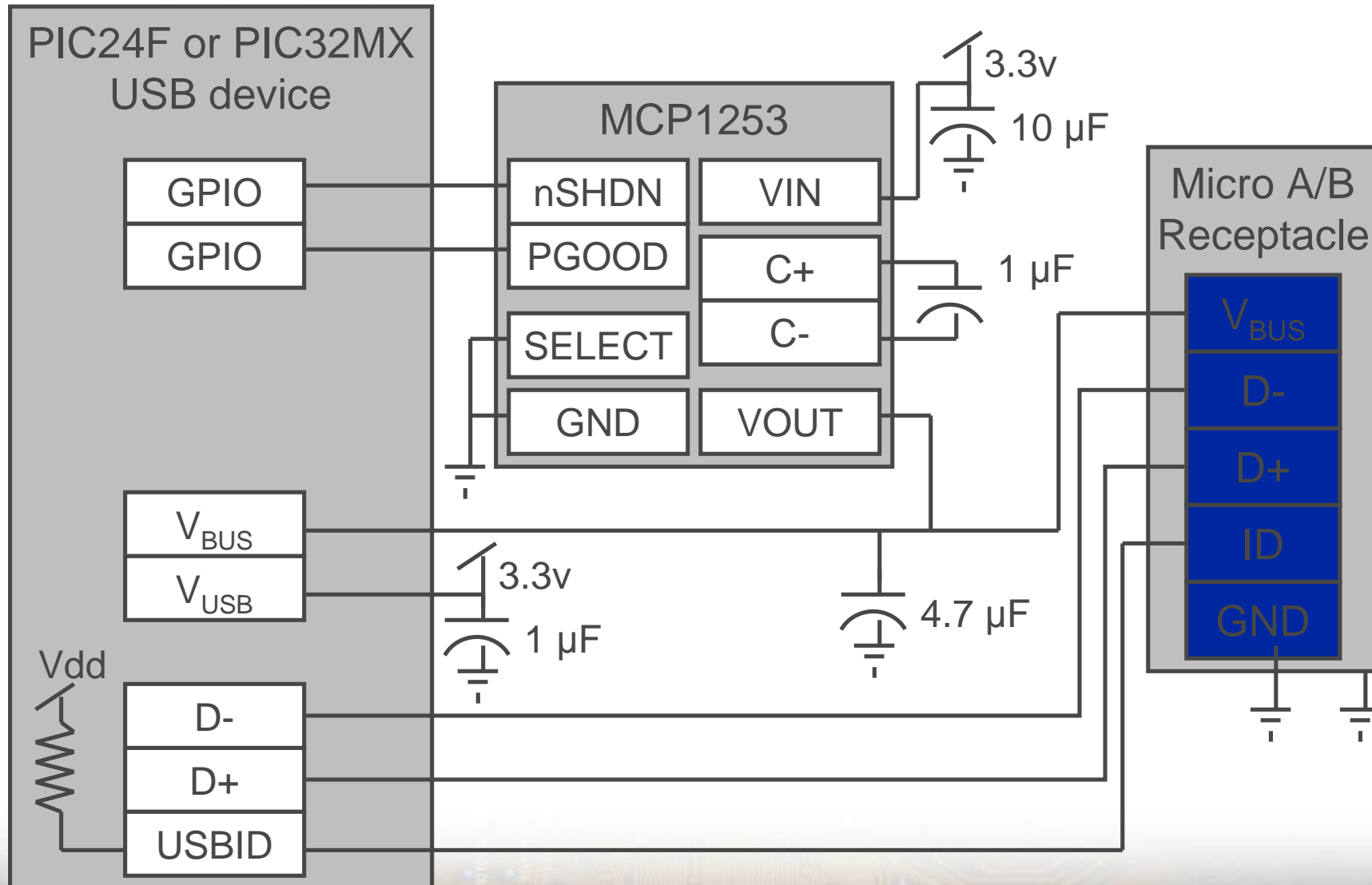
# USB Device (Peripheral) Example Circuit



# Embedded Host Example Circuit

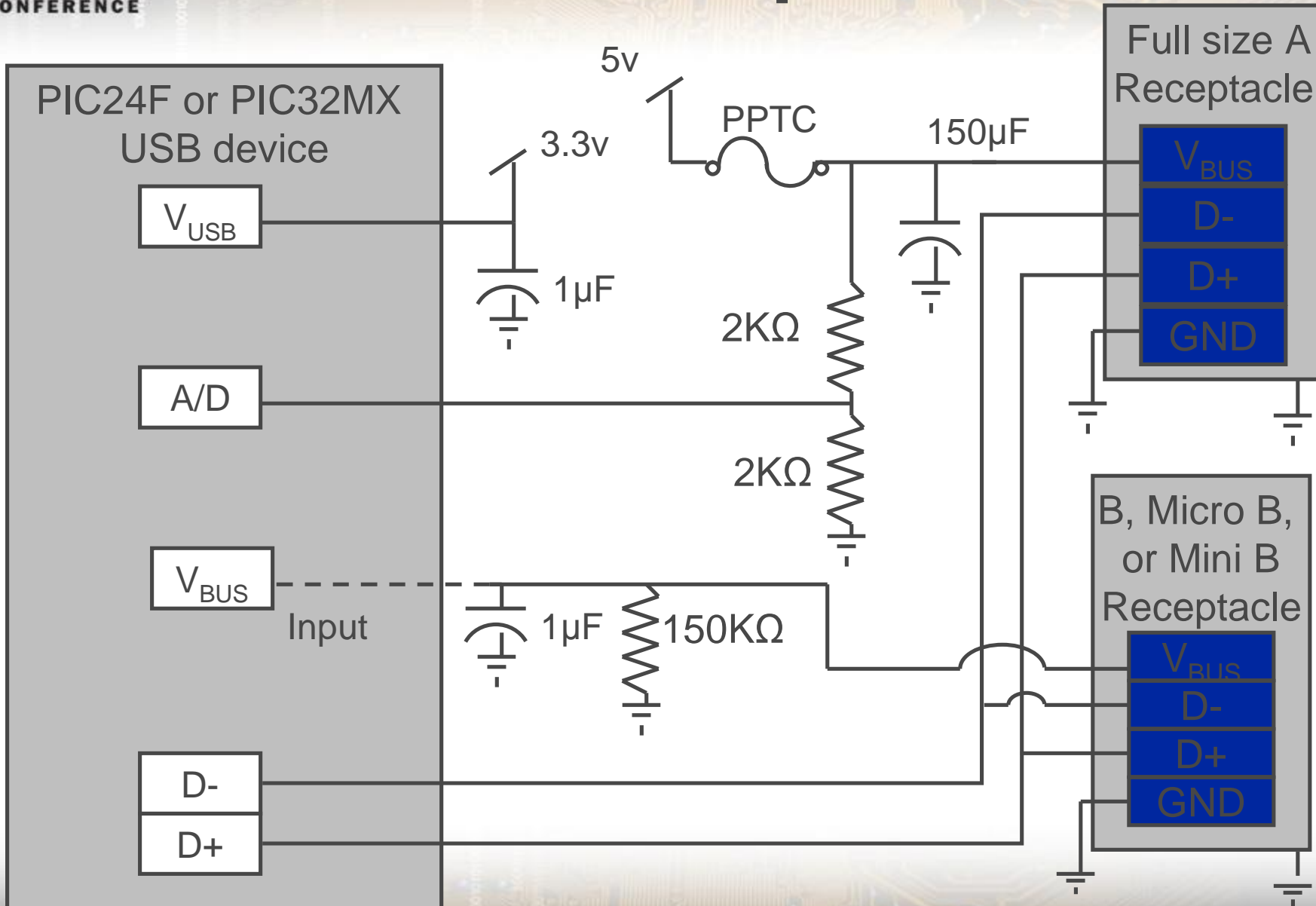


# OTG Example Circuit

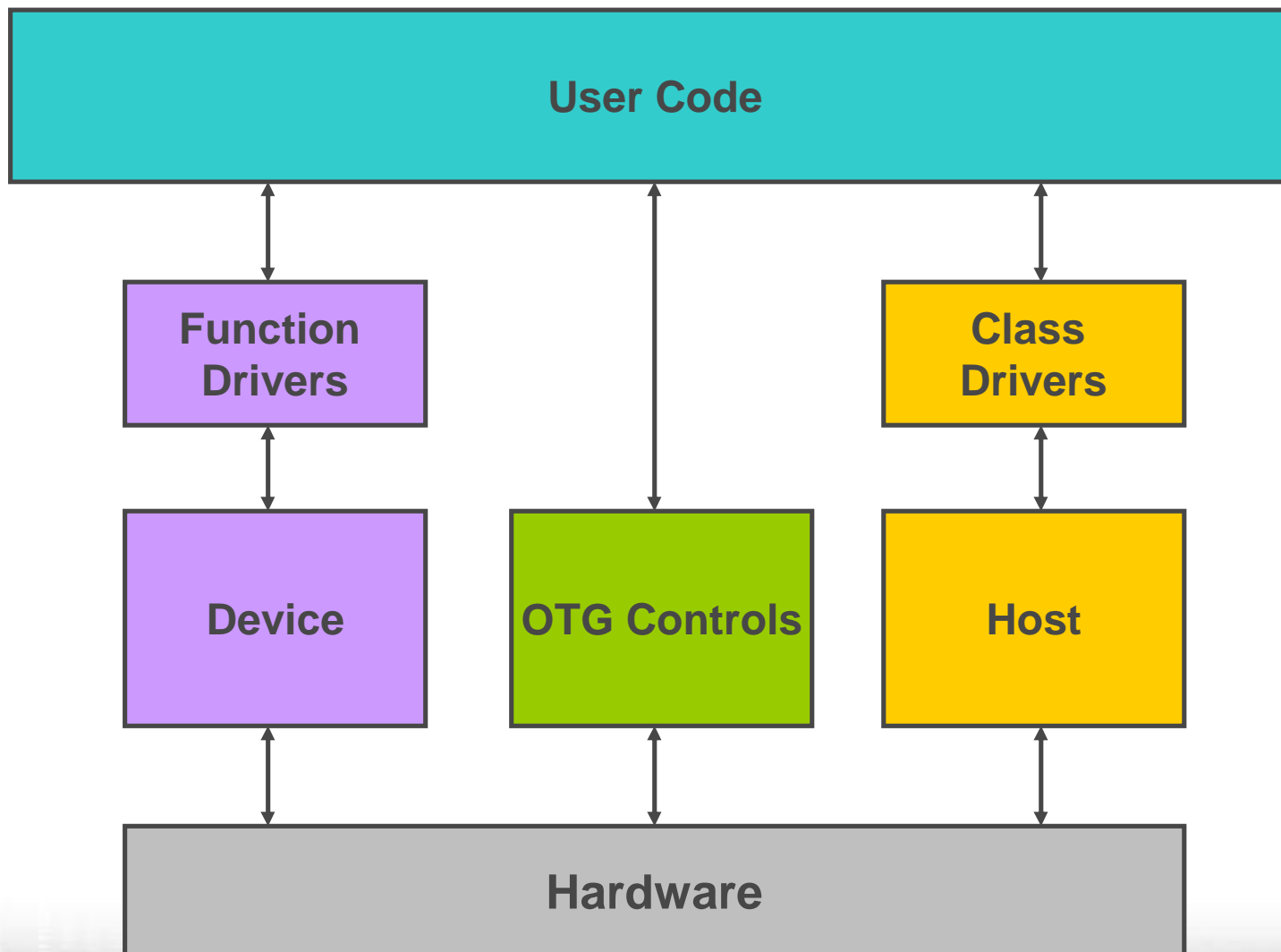




# DRD Example Circuit



# Структура ПО



# VID & PID

- | **Vendor ID (VID): 16-bit number**
  - | Required to market your product
  - | <http://www.usb.org/developers/vendor>
  - | **USD \$2,000**
  - | **Technical & Legal trouble if not using an approved VID**
- | **Product ID (PID): 16-bit number**
  - | **Microchip's Sub-licensing Program**
- | **Every product line is required to have a unique combination of VID and PID**

# USB Compliance

## Compliance Testing

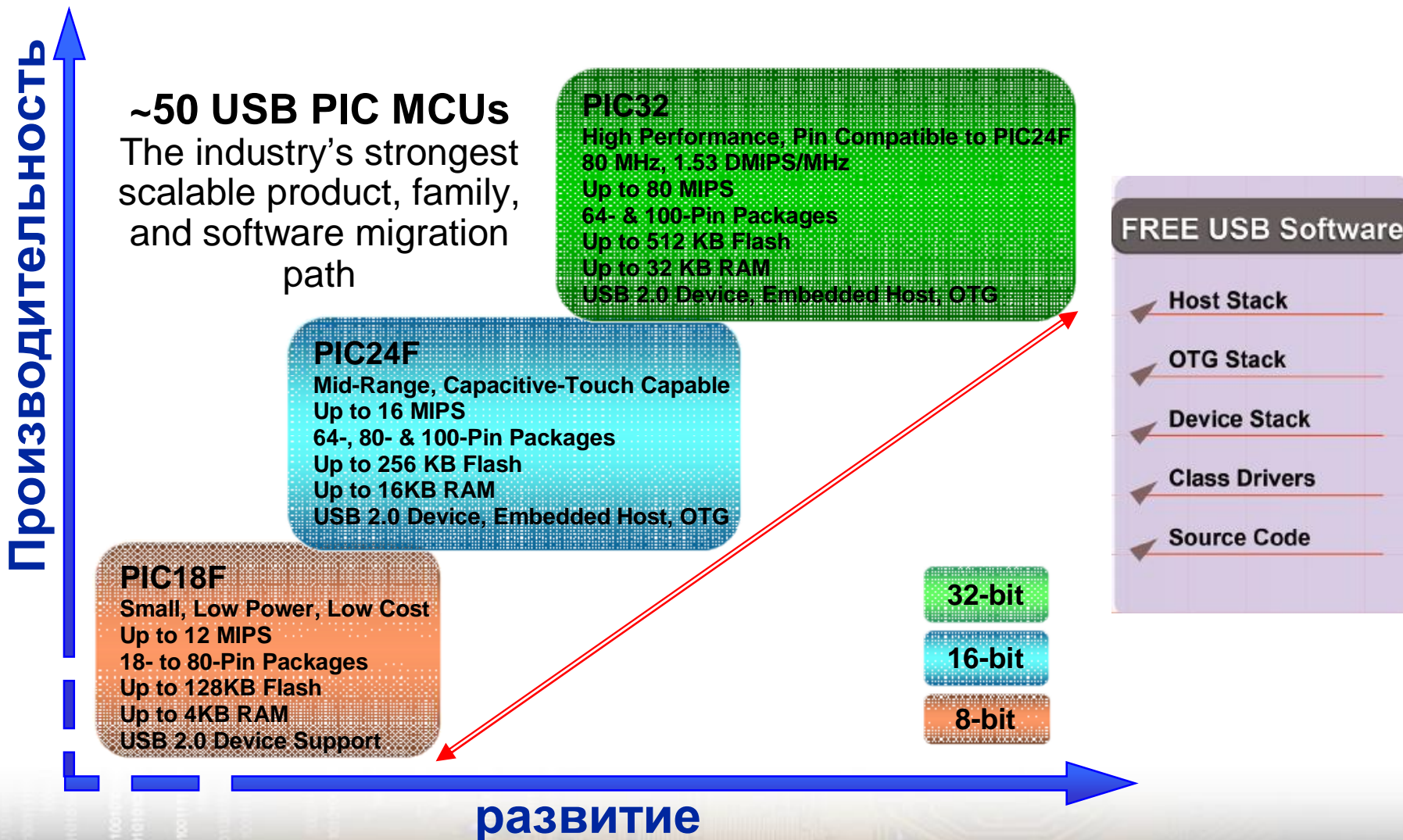
- Must pass to use USB logo
- Test fee: USD ~\$1,500



- Tests device for conformance to USB Device Framework and Class standard control requests
  - USB Protocol Analyzer
  - “USBCV” USB Command Verifier
  - [www.usb.org/developers/tools](http://www.usb.org/developers/tools)
- Electrical Signal Quality
- Power Management



# PIC с USB







# USB Microcontroller Portfolio

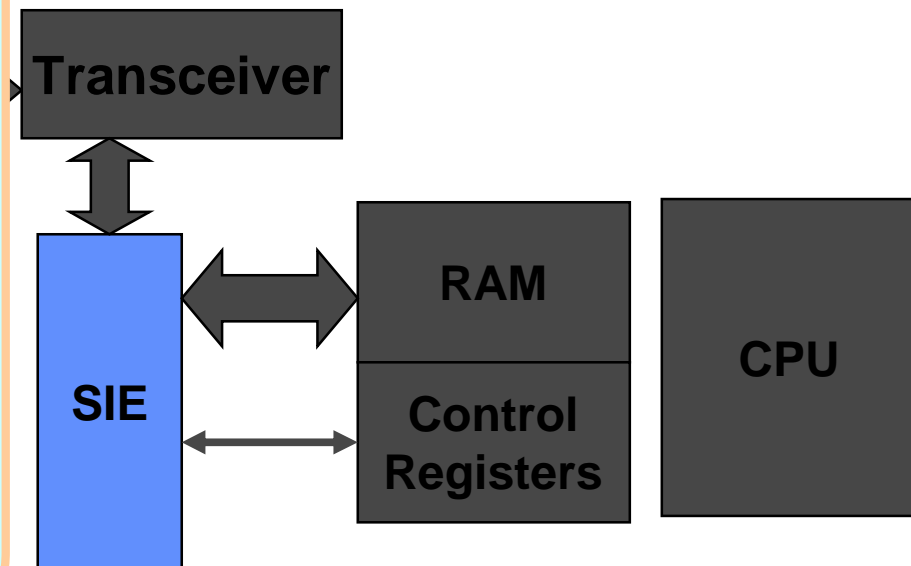
MASTERS CONFERENCE	PIC18F14K50	PIC18F4450 PIC18F4550 PIC18F4553	PIC18F87J50	PIC24FJ256GB1xx	PIC32MX4xxF512
<b>Core</b>	8 bit	8 bit	8 bit	16 Bit	32 Bit
<b>USB</b>	USB 2.0 device	USB 2.0 device	USB 2.0 device	USB 2.0 device, embedded host, dual role, OTG	USB 2.0 device, embedded host, dual role, OTG
<b>Flash</b>	16K bytes	up to 32K bytes	128K bytes	256K bytes	512K bytes
<b>RAM</b>	768 bytes	up to 2048 bytes	3904 bytes	16K Bytes	32K Bytes
<b>mTouch™ support</b>	yes	yes, external	yes, external	yes CTMU	yes, external
<b>UARTs</b>	1	1	2	4	2
<b>SPI</b>	1	1	1	3	2
<b>I<sup>2</sup>C™</b>	1	1	1	3	2
<b>Peripheral pin select</b>	no	no	no	yes	no
<b>ADC</b>	10 bit, 9 channel	10 bit, 10 and 13 ch 12 bit, 10 and 13 ch	10 bit, 8 and 12 channel	10 bit, 16 channel	10 bit, 16 channel
<b>RTCC</b>	software	software	software	yes	yes
<b>Parallel Master Port</b>	no	no	yes	yes	yes
<b>Analog comparators</b>	2	2	2	3	2
<b>Free sw stacks</b>	yes	yes	yes	yes	yes
<b>Free class drivers</b>	yes	yes	yes	yes	yes
<b>scalable development environment</b>	yes	yes	yes	yes	yes
<b>Packages</b>	20 pin	28,40,44 pin	60,80 pin	64,80,100 pin	64,100 pin

# Serial Interface Engine

## *SIE ...*

- | Прием и передача последовательных данных
- | Кодирование NRZI
- | Работа с битами
- | Подсчет CRC
- | Работа с событиями и прерываниями
- | Обработка USB транзакций
- | Поддержка соединения

USB PIC<sup>®</sup> MCU





# PICDEM™ Full Speed USB Demo Kit

- | **PIC18F4550**
- | **DM163025**



# PIC18FxxJ50 Full Speed USB Plug-In Module (PIM)

- | PIC18F87J50 или PIC18F46J50
- | Можно подключить к PICDEM™ HPC Explorer Board
- | Может работать автономно
- | MA180021 - PIC18F87J50 FS USB PIM
- | MA180024 - PIC18F46J50 FS USB PIM
- | DM183022 - PICDEM HPC EXPLORER BOARD





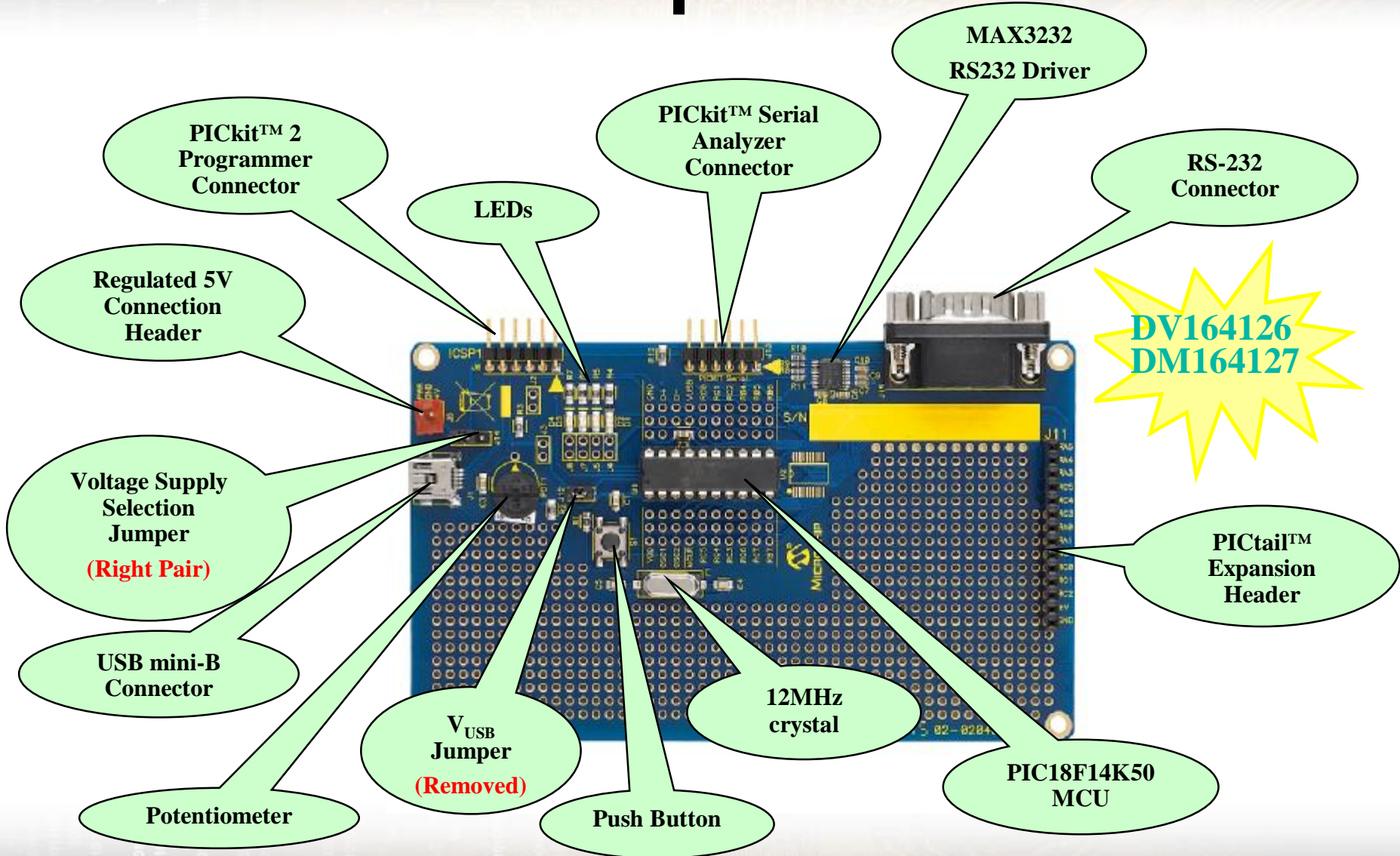
# Low Pin Count USB Development Kit

- | **PIC18F14K50**
- | **Примеры и быстрое начало работы:**
  - | Преобразователь RS232 - USB
  - | Клавиатура, мышь и т.д.
- | **DV164126 (w/ PICkit™ 2)**
- | **DM164127**





# Low Pin Count USB Development Board





# PIC18F14K50 Low Pin Count USB MCU

## Features

- | **8-bit, 12 MIPS**
- | **Voltage 1.8 to 5.5V**
- | **16 KB Flash**
- | **512 Bytes SRAM**
- | **256 Bytes DPRAM**
- | **256 Bytes EEPROM**
- | **20 pins**

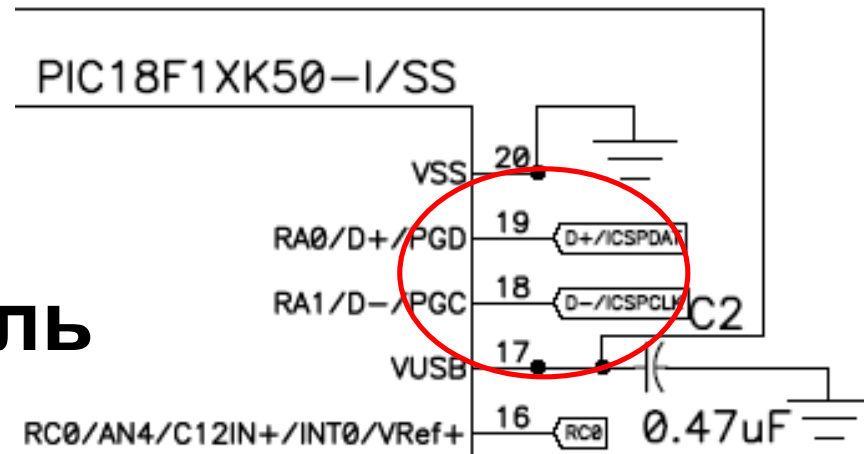
## Peripherals

- | **ECCP**
- | **SPI/I<sup>2</sup>C™**
- | **EUSART**
- | **9 ch. x 10-bit A/D**
- | **1x 8-bit, 3x 16-bit timers**
- | **2x Comparators**
- | **USB 2.0 Full Speed**

**Optimized for protocol-conversion applications**

# PIC18F14K50: D+/D- и PGC/PGD

- Отключайте USB при программировании
- Для отладки доступен специальный отладочный модуль

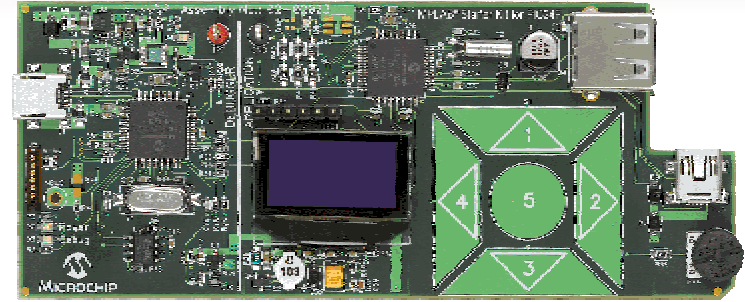




# 16-/32-bit USB Development Boards

- | **PIC24F Starter Kit 1**

- | Part #: DM240011
- | PIC24FJ256GB110



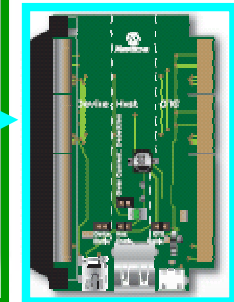
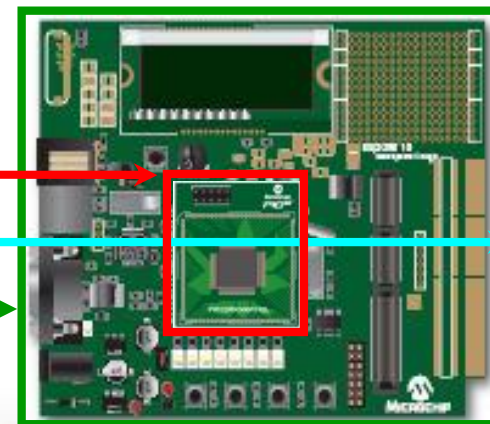
- | **PIC32 USB Starter Board**

- | Part #: DM320003
- | PIC32MX460F512



- | **Explorer 16 + USB PICtail™ Plus Daughter Board + USB PIMs**

- | Part #: MA320002/MA240014
- | Part #: AC164131
- | Part #: DM240001





# Microchip USB Frameworks

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

- | **MCHPFSUSB v2.4a**
  - | PIC18F & PIC24F USB MCUs (PIC32 Beta support)
  - | C18/C30(/C32) Compatible
  - | MPLAB® IDE Project Centric
- | **Device Stacks**
  - | HID, CDC, MSD, Custom
  - | Polling or Interrupt driven
- | **Embedded Host Stack**
  - | Polling or Event-driven Scheme
  - | PIC24F USB MCUs (PIC32MX460F512L Beta)
    - | API compatible with PIC32 USB Framework
- | **On-The-Go (OTG) Support**
  - | PIC24F only
- | **PIC32 USB Framework v1.04**
  - | PIC32 USB MCUs
  - | C32 Compatible
  - | MPLAB IDE Project Centric
- | **Device Stacks**
  - | HID, CDC, MSD, Custom
  - | Polling or Interrupt driven
- | **Embedded Host Stack**
  - | Polling or Event-driven Scheme
  - | API compatible with MCHPFSUSB stack

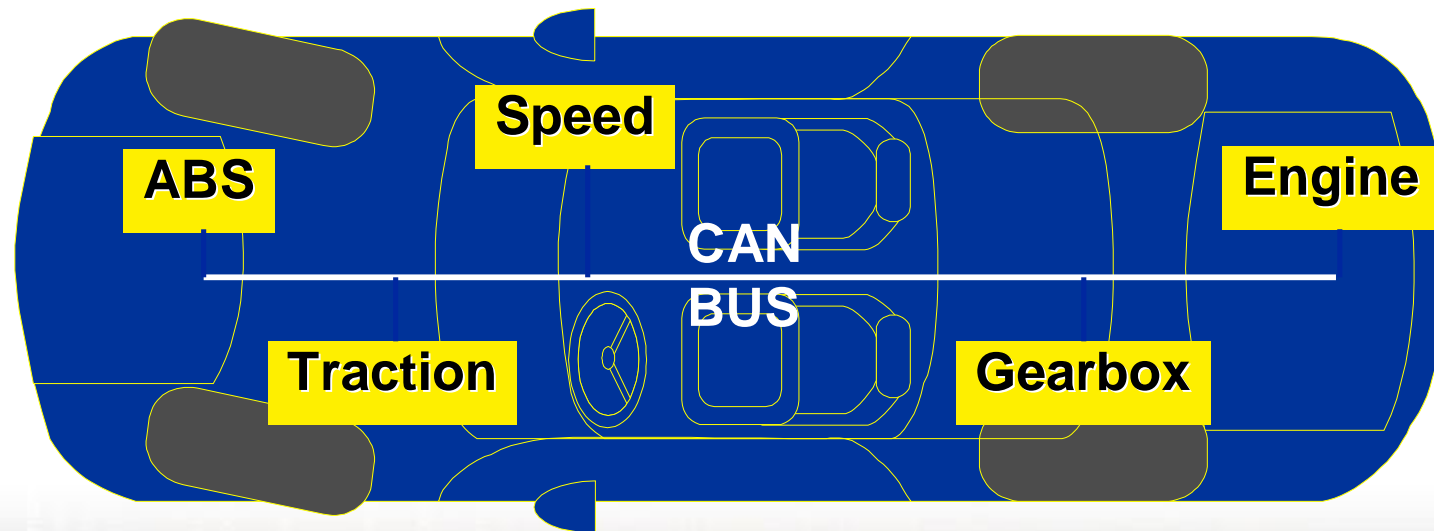




# CAN

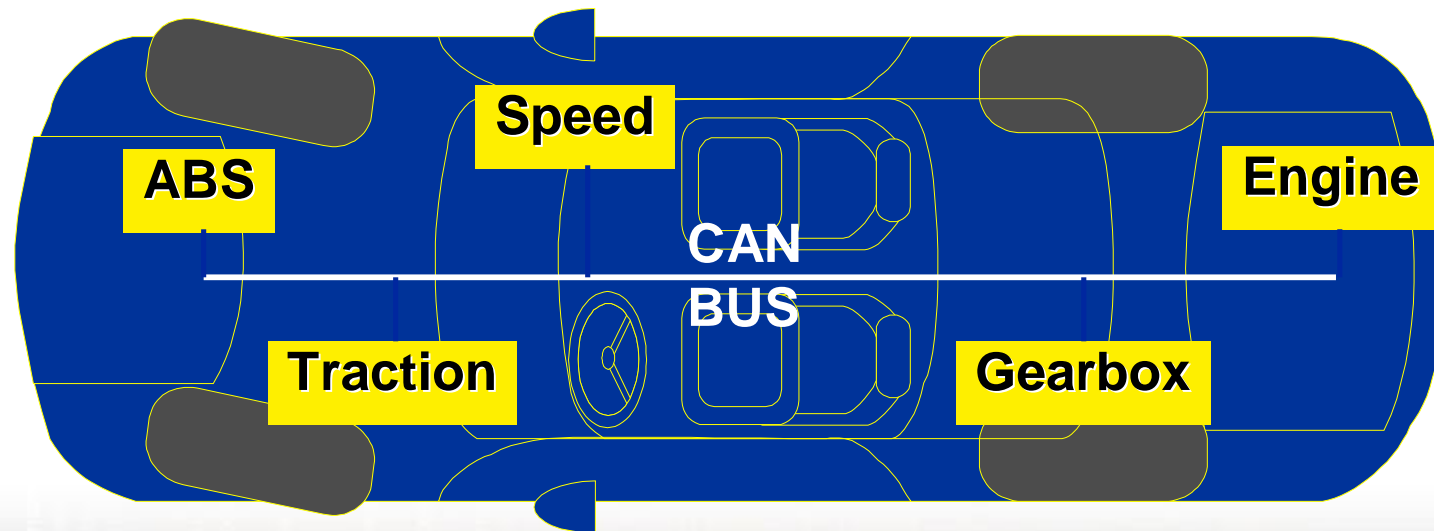
# Что такое CAN

- | Последовательный протокол
- | Все узлы постоянно подключены к общей шине
- | Все узлы работают на одной скорости обмена
- | Любой узел может передавать и принимать сообщения



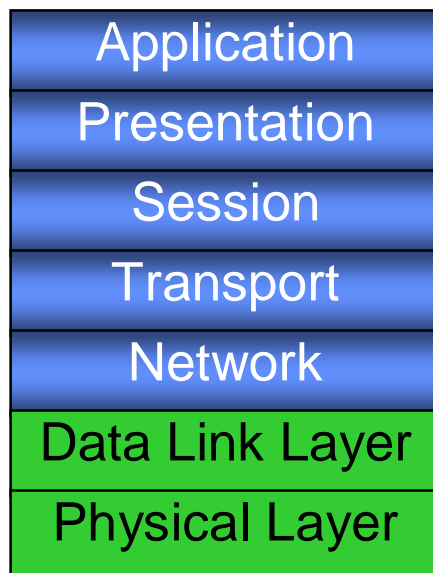
# Что такое CAN

- | Каждый узел имеет свою функцию и идентифицируется с этой функцией
- | На этапе разработки указывается идентификаторы и типы сообщений для каждого узла
- | ПО готовит сообщения к передачи и обрабатывает принятые сообщения



# Сетевая модель CAN

## OSI Reference Layers



DATA LINK LAYER

PHYSICAL LAYER

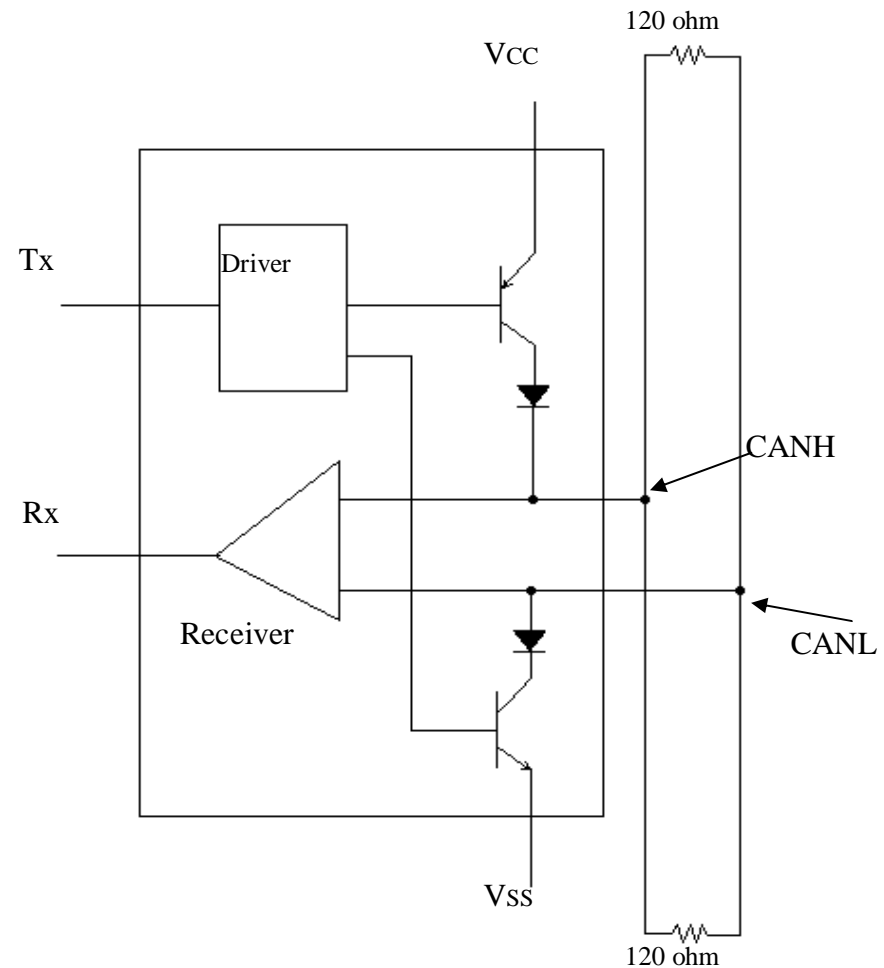
# Физический уровень CAN

- | **CAN High Speed**
  - | ISO11898-2
  - | До 1 Мбит/с
- | **CAN Low Speed (Fault Tolerant)**
  - | ISO11898-3
  - | До 125 кбит/с

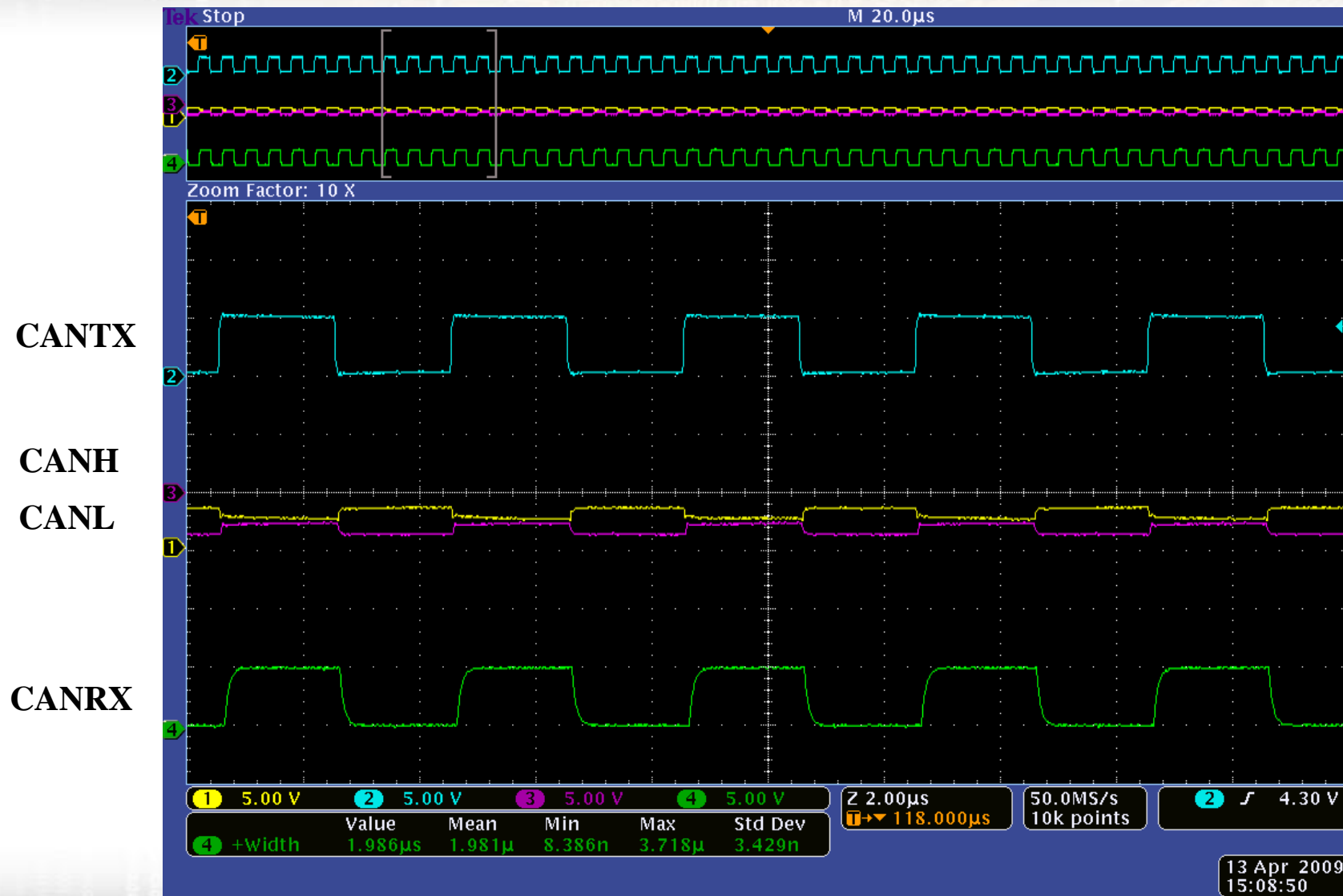


# CAN High Speed

- | **CAN-трансивер (MCP2551)**
- | **Питание 4.5..5.5 В**
- | **Tx и Rx – сигналы микроконтроллера**
- | **CANH и CANL – дифференциальные линии шины: от 0 до 3В**
  - | DV > 1.0 В доминантный
  - | DV < 0.5 В рецессивный
- | **Устойчивая работа при наведенном напряжении До ±40 В**
- | **Выдерживает наводки до ±200 В**
- | **Макс. длина кабеля:**
  - | 40 м для 1 Мбит/с
  - | 1 км для 50 кбит/с



# Сигналы на шине



# Основные особенности CAN

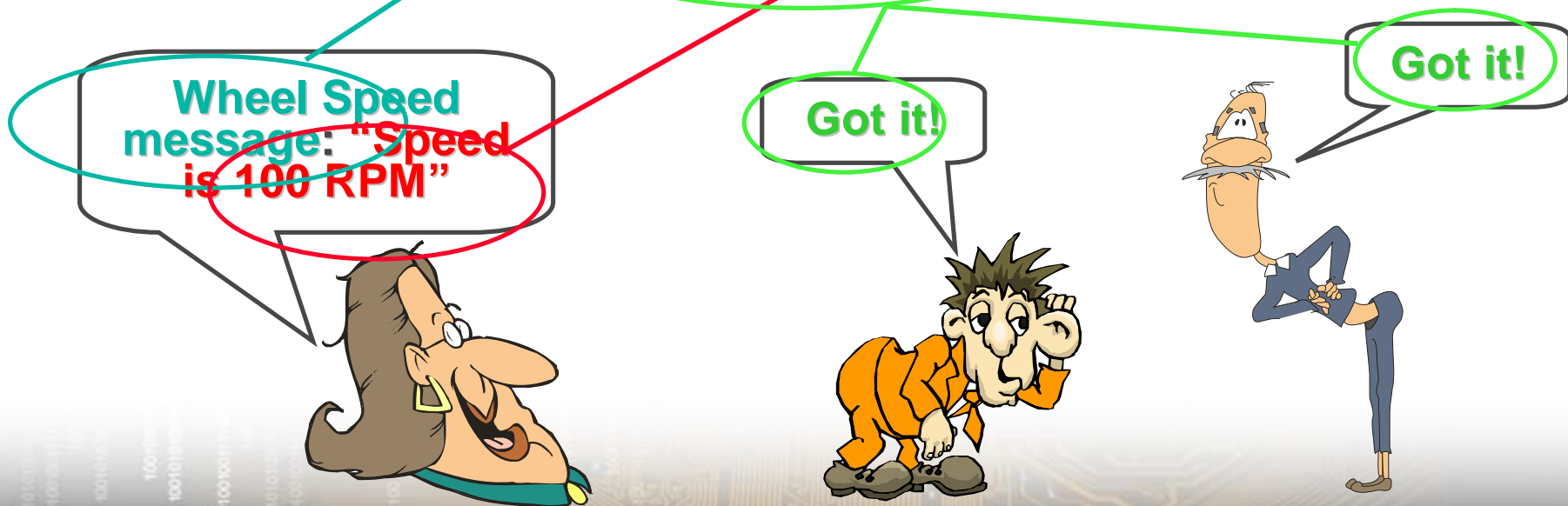
- | **Ориентирована на сообщения, а не на адрес**
  - | В шину посылается сообщение
  - | Иногда узлы могут сами запросить сообщение у других узлов
- | **Быстрое, надежное соединение**
- | **Carrier Sense Multiple Access and Collision Detection with Collision Resolution (CSMA/CD-CR)**

# CSMA/CD-CR

- | **Collision Resolution (CR)** –  
неразрушающий арбитраж на  
битовом уровне
  - | Сообщения не повреждаются во время коллизии
  - | Арбитраж не повреждает более приоритетное сообщение
  - | Сообщение, поврежденное при арбитраже, автоматически пересылается

# Сообщения CAN

- Только одно сообщение в данный момент времени
- Сообщение: “Идентификатор” + “Данные”
- Сообщение посылает всем на шине
- Все приемники, кто получил сообщение, этот прием *подтверждают*





# Формат сообщений

- | **Идентификатор (Арбитр)** в начале сообщения
  - | Два формата, стандартный и расширенный
- | **Данные** могут быть от 0 до 8 байт
- | Кроме этого, есть поля **Start**, **End**, **CRC**, **Control**, и **Acknowledge**

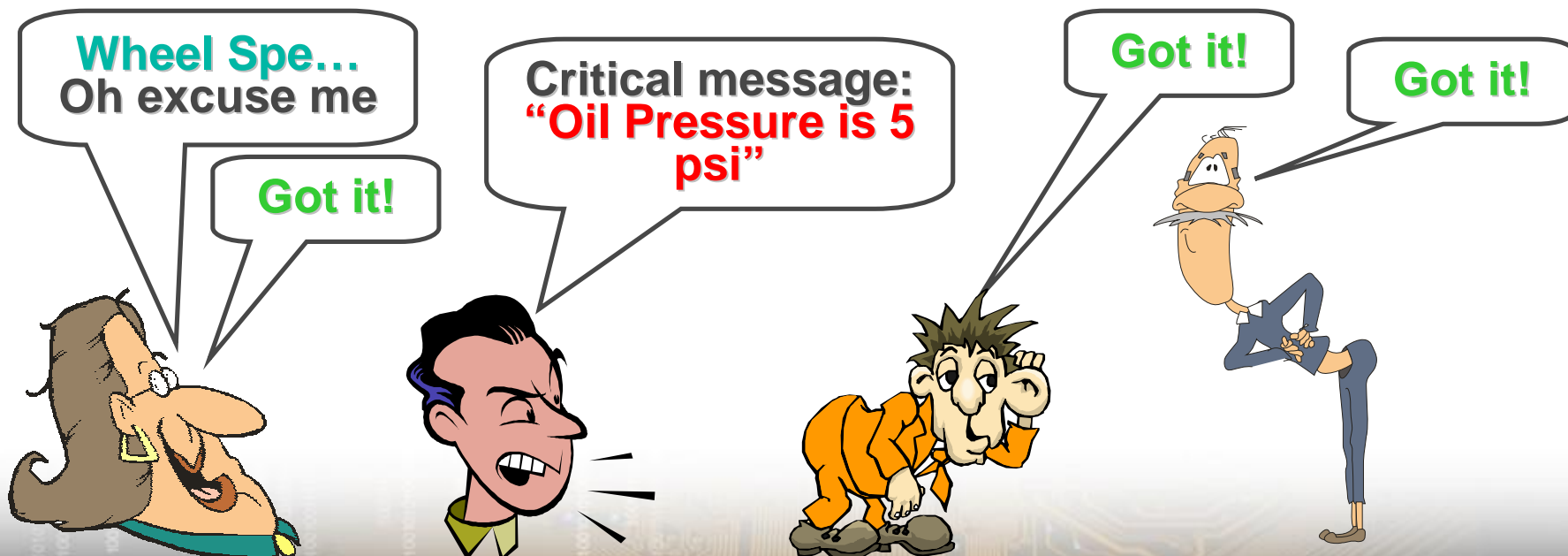


## Ориентирована на сообщения

- | **Сообщения не передаются по адресу**
- | **Главное в сообщении – приоритет и данные**
- | **Чем ниже идентификатор, тем выше приоритет**
- | **Все узлы принимают каждое сообщение и посылают подтверждение. Это – аппаратная функция модуля CAN**
- | **Только после этого конкретный узел решает, что делать с данными**

# Ориентирована на сообщения

- Узлы ожидают перерывы в передаче, чтобы начать свою
- Если два узла начнут передавать одновременно, то сработает арбитраж по идентификатору



# Арбитраж

- Два узла продолжают передачу до несовпадения в битах
- «Ноль» на шине имеет приоритет перед «единицей»

**Engine Control**

1	9	6
0	0	1
1	1	0
0	0	1
0	1	1
0	1	0

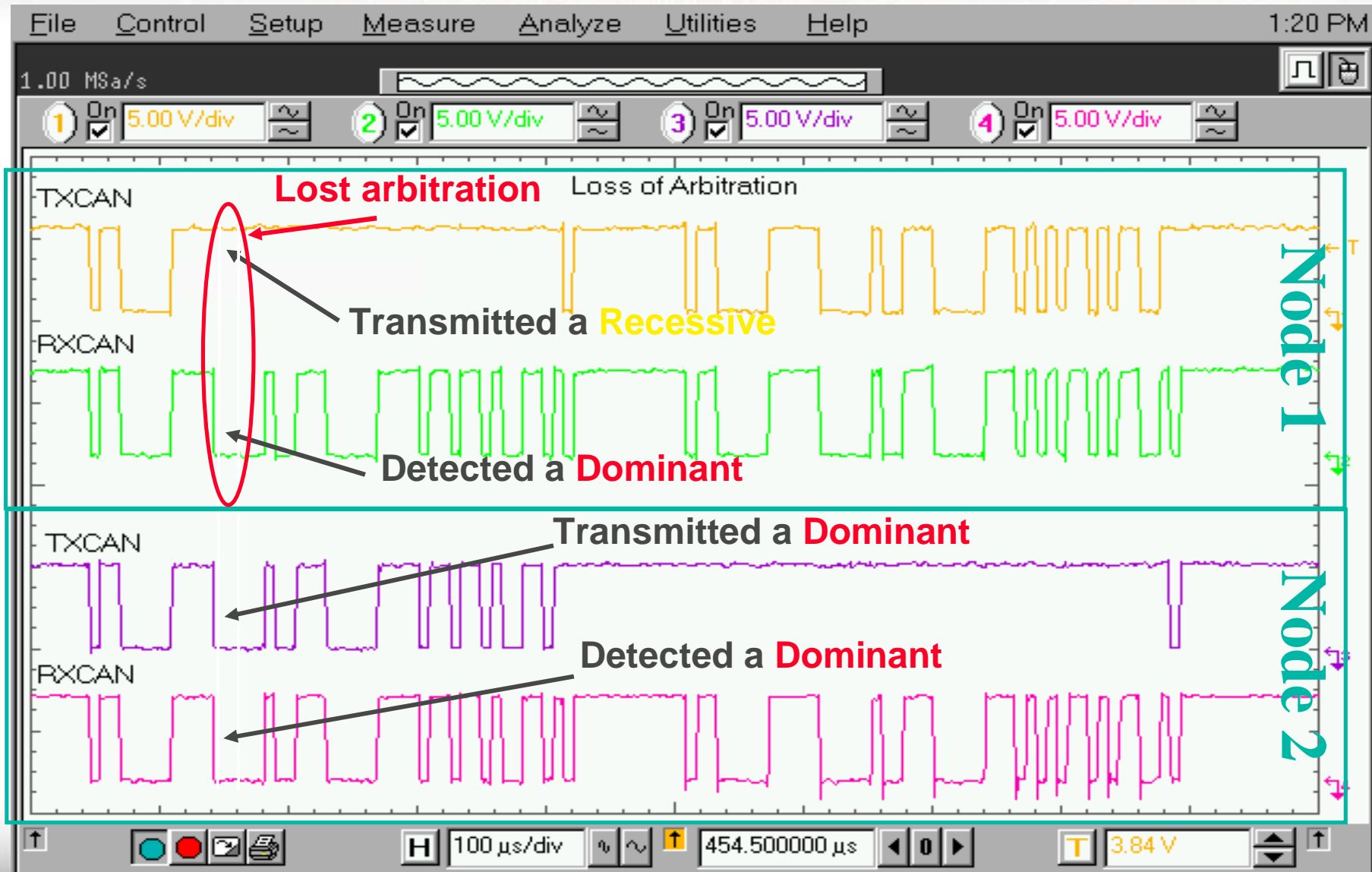
“Critical Message / Engine = 196h”

**Wheel Speed**

1	9	E
0	0	1
1	1	0
0	0	1
1	1	0
0	0	1
0	1	1

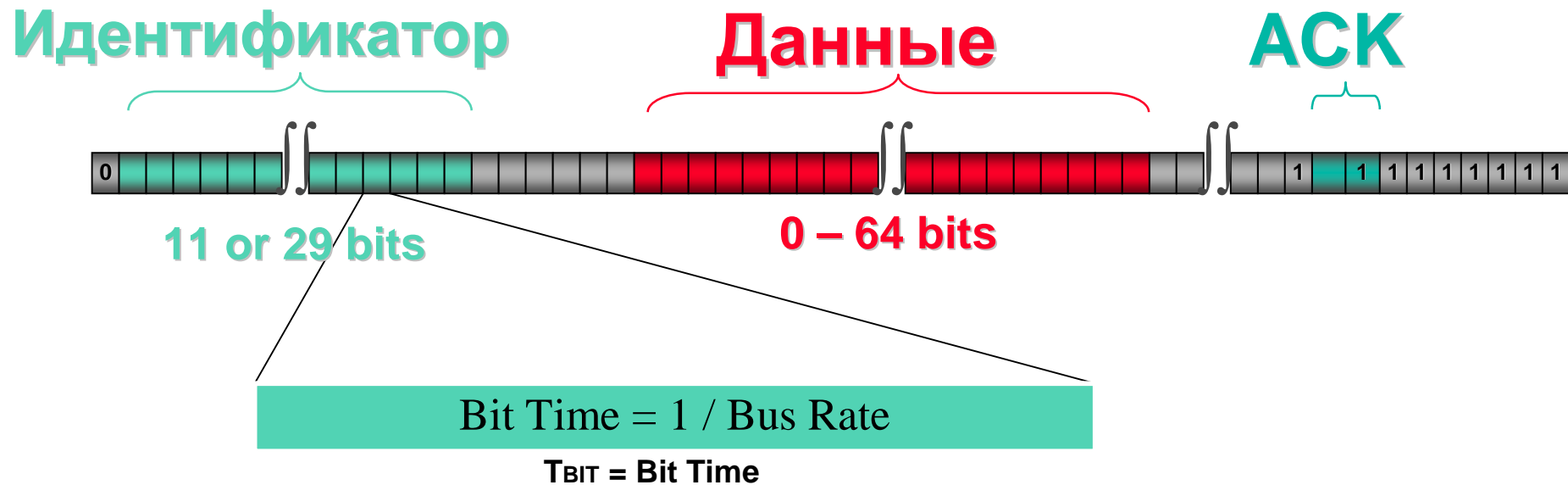
“Important Message / Wheel Speed = 19Eh”

# Пример арбитража





# Временные интервалы



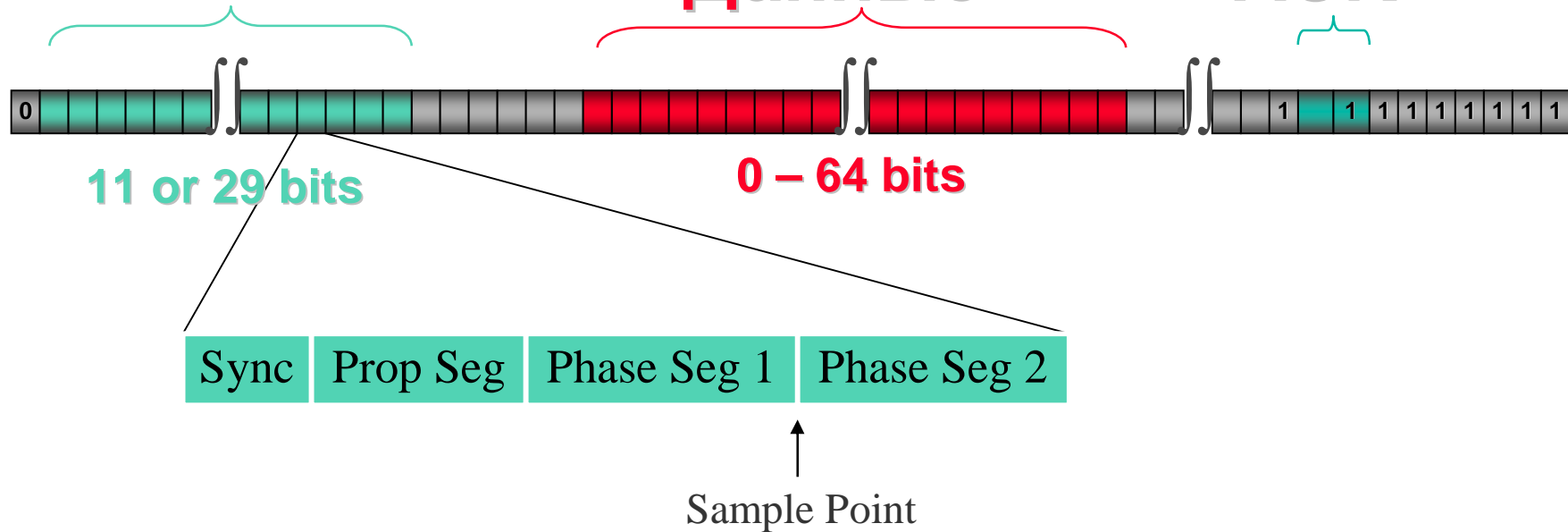
- | **Скорость шины определяется исходя из времени передачи одного бита**
  - | 1 Мбит/с -> 1 мкс битовый интервал

# Временные интервалы

Идентификатор

Данные

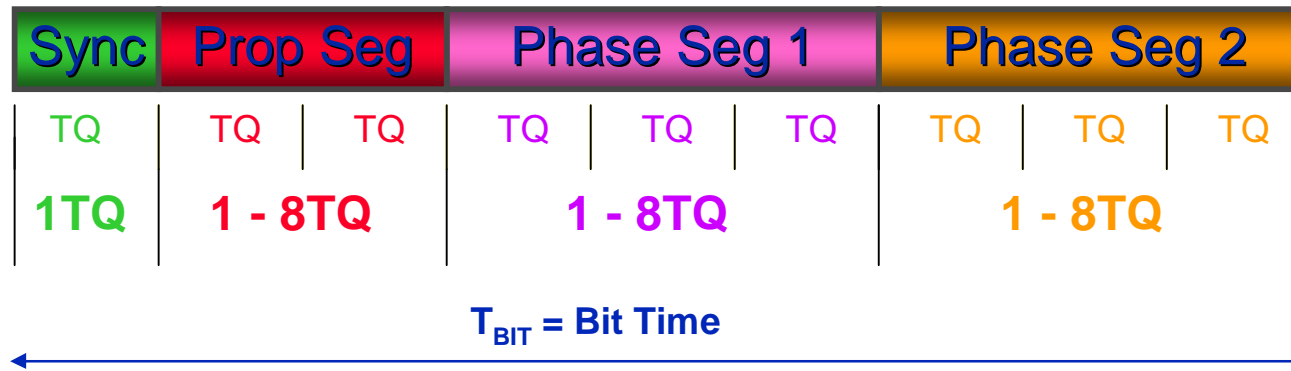
АСК



Бит на шине может делиться на 4 сегмента

# Временные интервалы

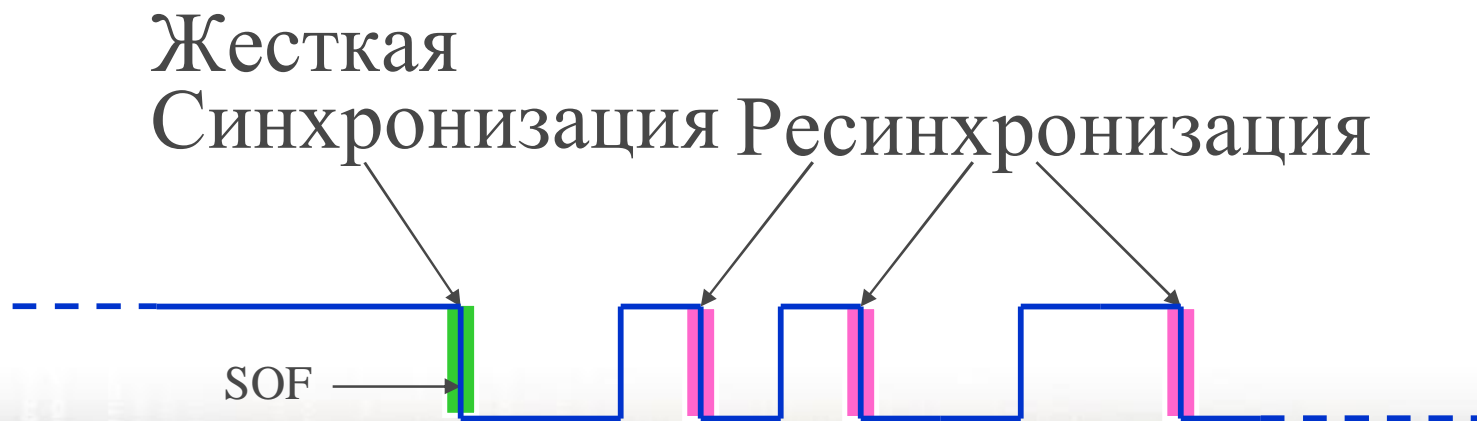
- Каждый сегмент состоит из временных квантов (Time Quanta, TQ)



- $TQ = 2 \cdot (BRP) \cdot T_{osc}$
- Baud Rate Prescaler (BRP):
  - Min = 1:1, Max = 1:64
- Битовый интервал может быть от 8 до 25 TQ

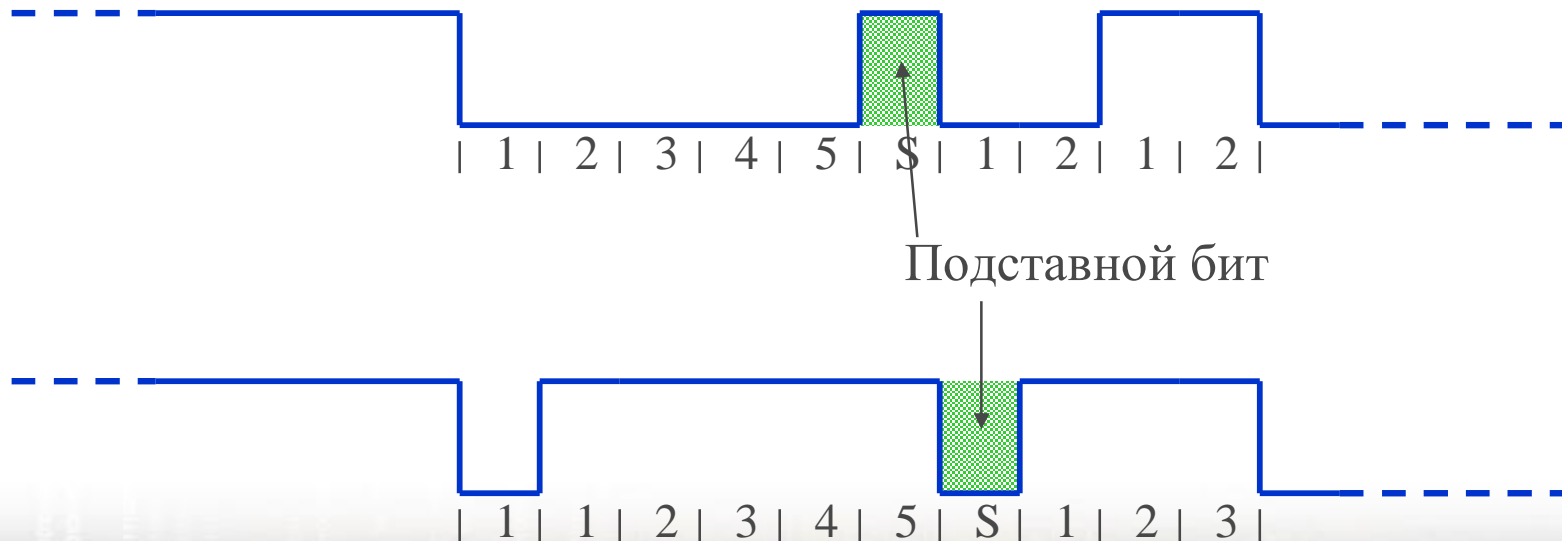
# Синхронизация

- Тактов в битовом потоке нет
- Приемники синхронизируются на переходе из рецессивного в доминантный
  - Жесткая синхронизация в начале кадра
  - Ресинхронизация происходит при переходе бита из рецессивного в доминантный (1-в-0)



# Подстановка битов

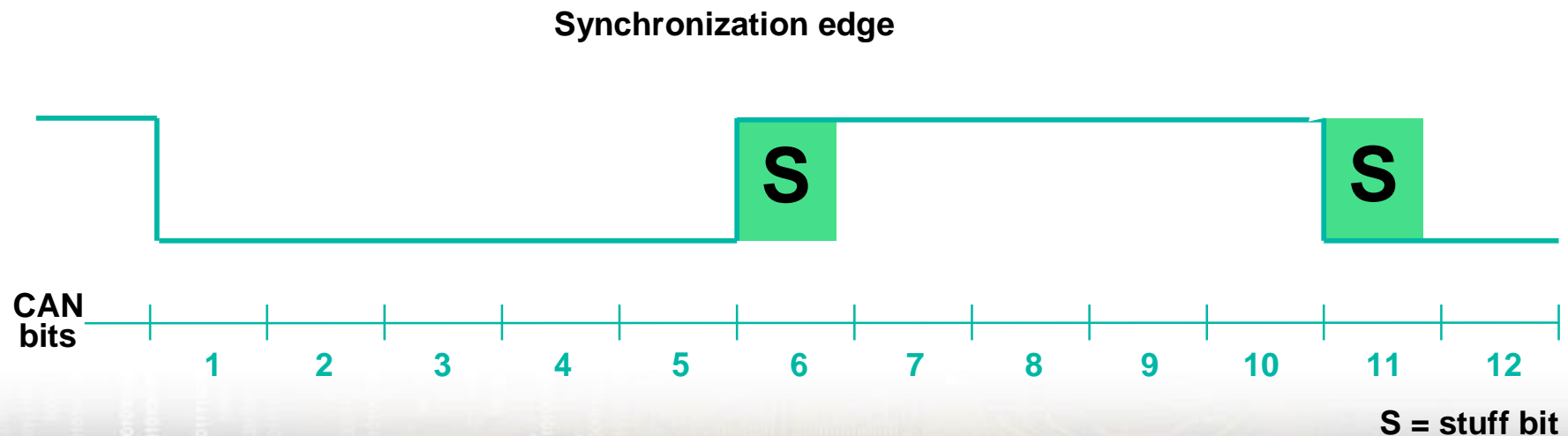
- Подстановка битов гарантирует наличие переходов для ресинхронизации
- Подставной бит ставится через каждые пять битов в потоке





# Подстановка битов

- Максимальный интервал между ресинхронизациями – 10 бит



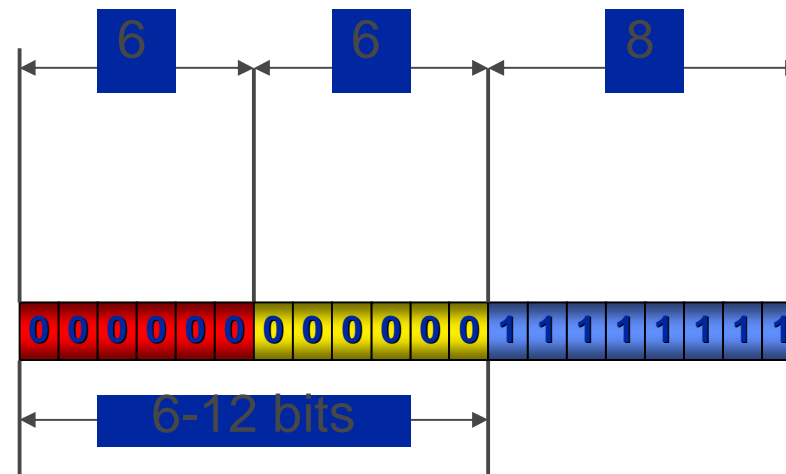
# Обработка ошибок

- | **Несколько типов ошибок**
  - | Целостность сообщений
- | **Работа над ошибками**
  - | Узлы CAN могут быть переведены из режима нормальной передачи в режим отключения от сети на основе данных об ошибках
  - | Это решение об отключении ошибочных узлов позволяет защитить сеть от перегрузки

# Кадр с ошибкой

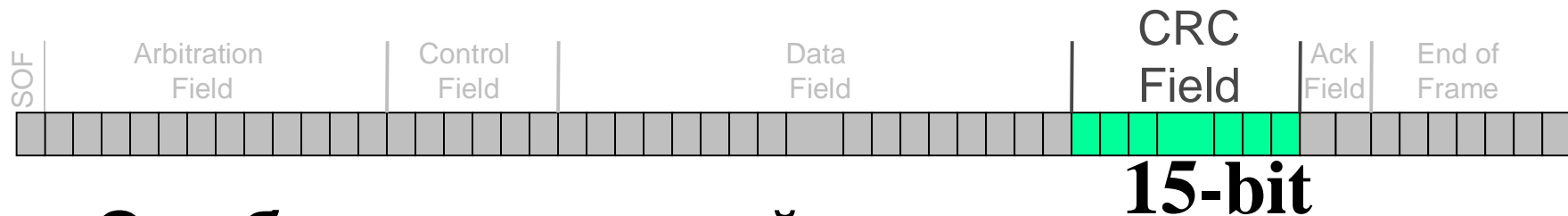
- Когда любой из узлов находит ошибки в сообщении, он посылает Кадр с ошибкой

Флаги    Повтор    Заполнитель



Суперпозиция  
флагов

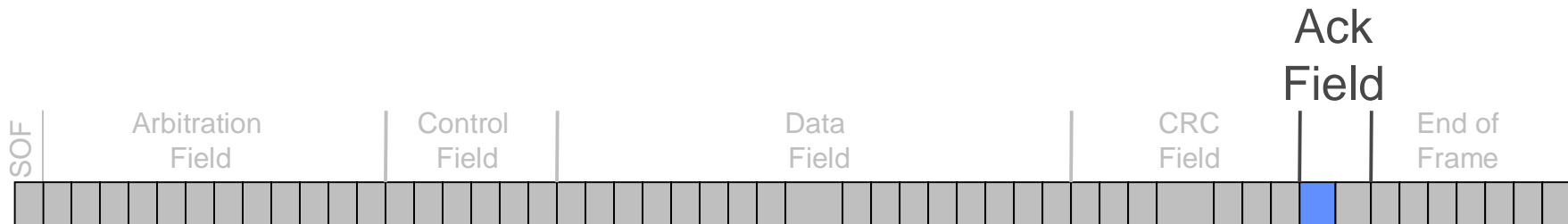
# Ошибка контрольной суммы



## I Ошибка контрольной суммы

- I 15-битный CRC
- I Все узлы при приеме вычисляют контрольную сумму и сравнивают ее с суммой, переданной в сообщении
- I Если суммы не совпадают, узлы генерируют кадр с ошибкой
- I Передающий узел получает этот кадр и повторяет сообщение

# Ошибка подтверждения

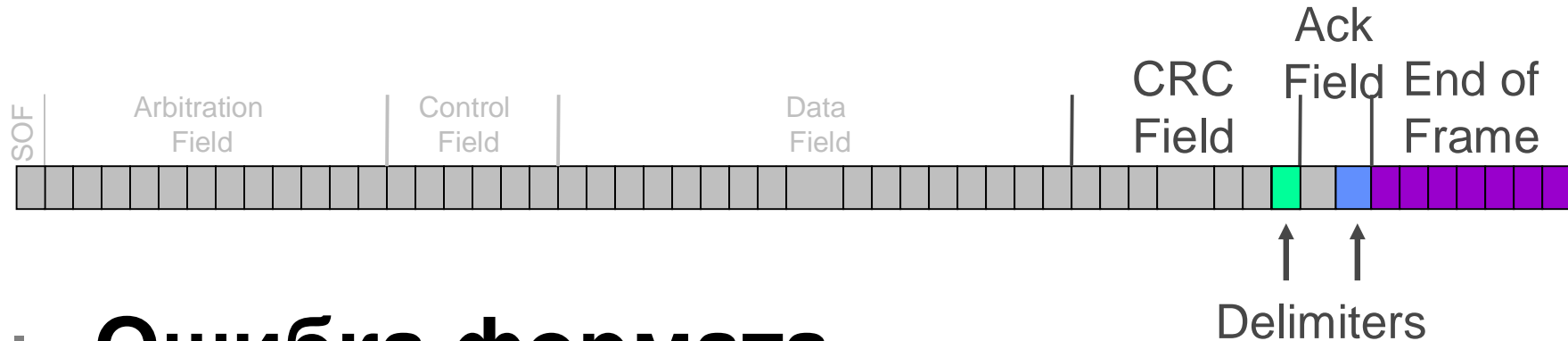


## Ошибка подтверждения

- И Передающий узел проверяет бит АСК, который он передает как рецессивный
- И Если бит стал доминантным, значит как минимум один узел принял сообщение правильно
- И Если нет, то генерируется Кадр с ошибкой и сообщение повторяется



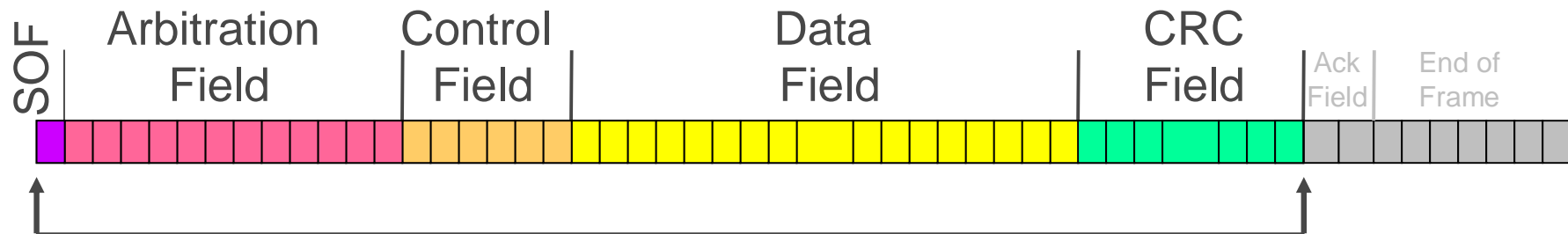
# Ошибка формата



## Ошибка формата

- Узел, обнаруживший доминантный бит в заполнителе после CRC, Ack или в поле Конец кадра (EOF) или сразу после него генерирует Кадр с ошибкой
- Сообщение пересылается заново

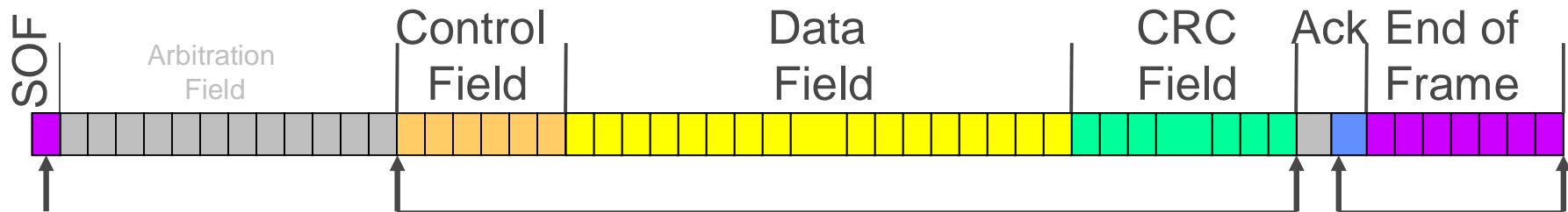
# Ошибка подстановки



## Ошибка подстановки бита

- Если принимается подряд 6 битов одной полярности в кадре, делается вывод об ошибке с подстановкой бита
- Генерируется Кадр с ошибкой и сообщение пересылается заново

# Битовая ошибка

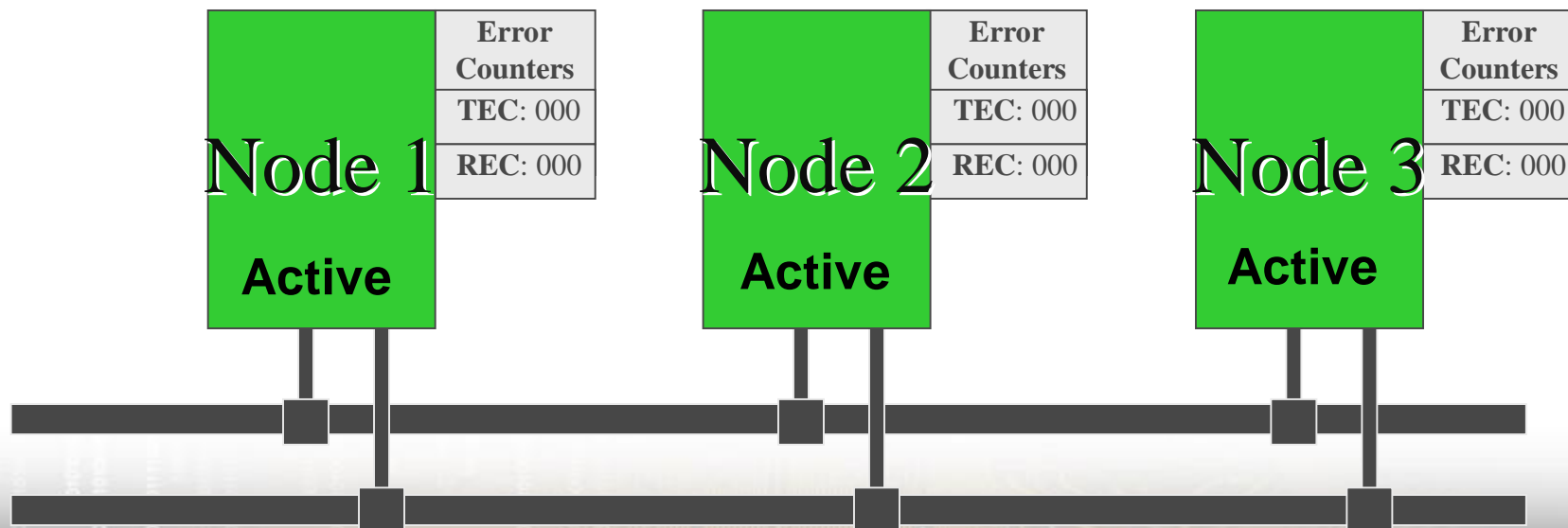


## Битовая ошибка

- Передатчик контролирует передаваемые данные. Если они не совпадают, делается вывод об ошибке и сообщение пересылается заново
- Исключения
  - Арбитраж
  - Бит подтверждения (Аск, в случае удачного приема)

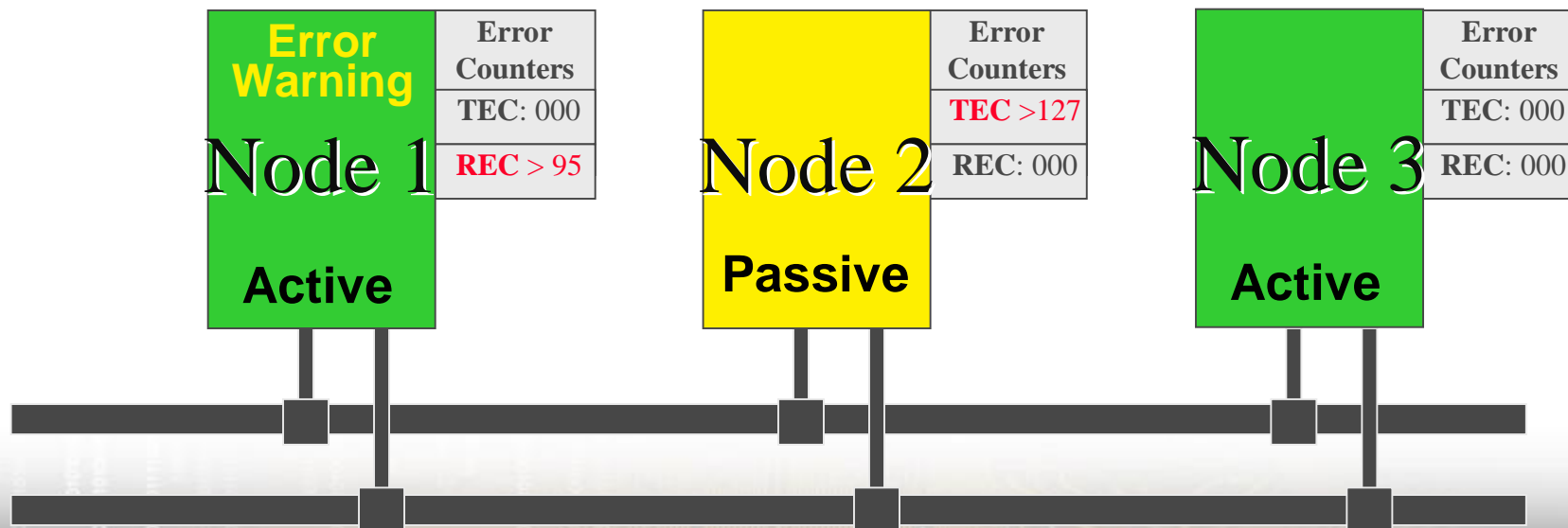
# Ограничения ошибок

- Определено три ошибочных состояния:  
Ошибочное активное, Ошибочное пассивное  
и Отключен
- Ошибочное активное – нормальный режим
  - Можно отсылать сообщения и Активные Кадры с ошибками



# Ограничения ошибок

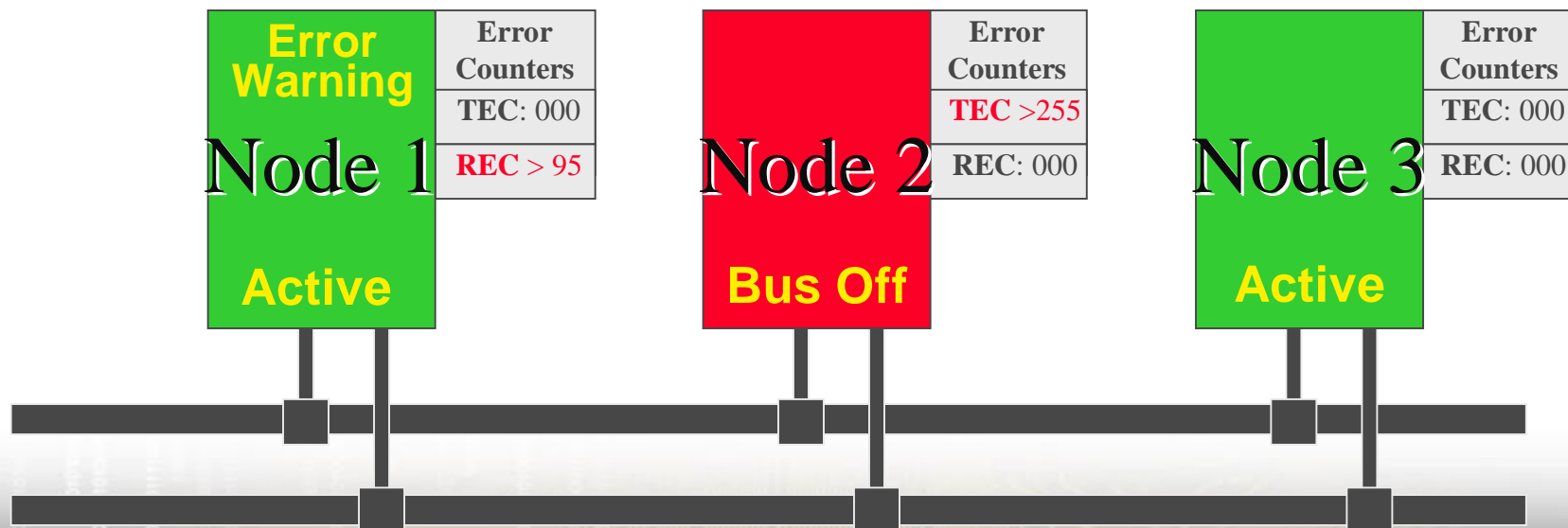
- I Когда любой из счетчиков достигает значения 95 – генерируется предупреждение (прерывание)
- I Когда 127 – узел переходит в Ошибочное пассивное состояние
  - I Может отсылать сообщения и Пассивные Кадры с Ошибками





# Ограничения ошибок

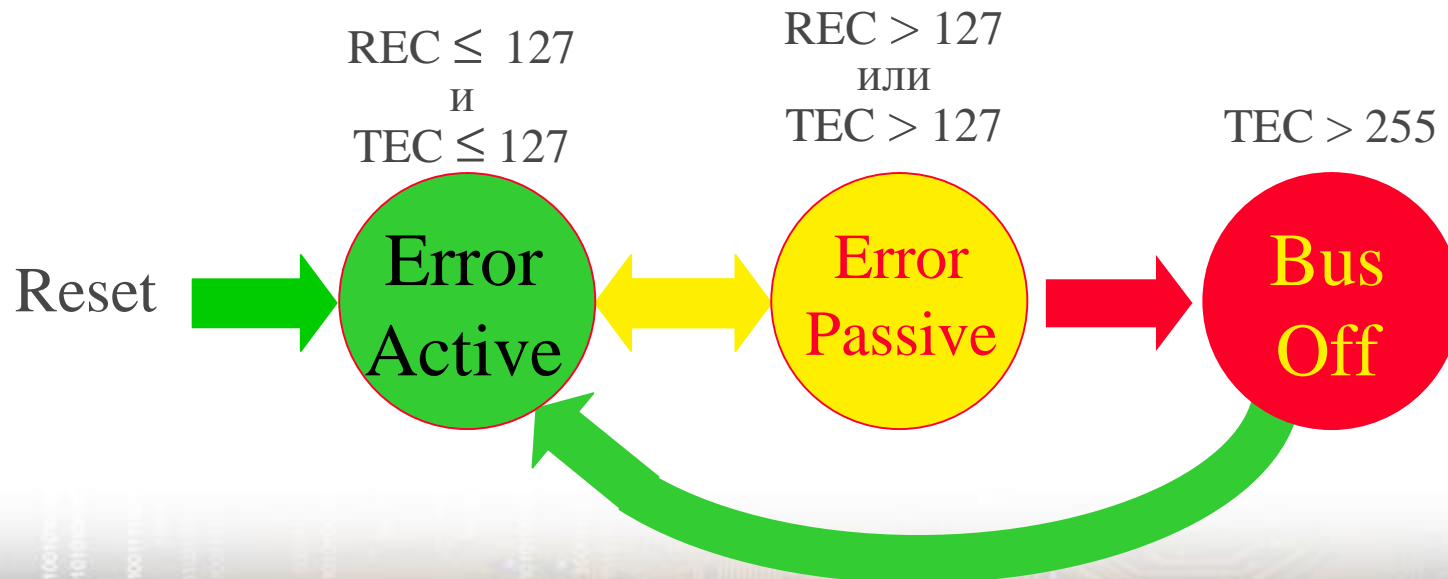
- Когда счетчик переданных ошибок становится больше 255, узел переходит в состояние Отключен
  - Узел ничего не передает



# Выход из состояния Отключен

## I 2 пути:

1. Переключиться в конфигурационный режим
2. Получить по шине 128 раз по 11 рецессивных бит (длинный простой шины) или 128 правильных сообщений, или комбинацию этих событий



# Модуль ECAN™

- **ECAN™ (Enhanced Controller Area Network) – обозначение нового модуля CAN и его особенностей, встроенного в PIC**

# Модуль ECAN™

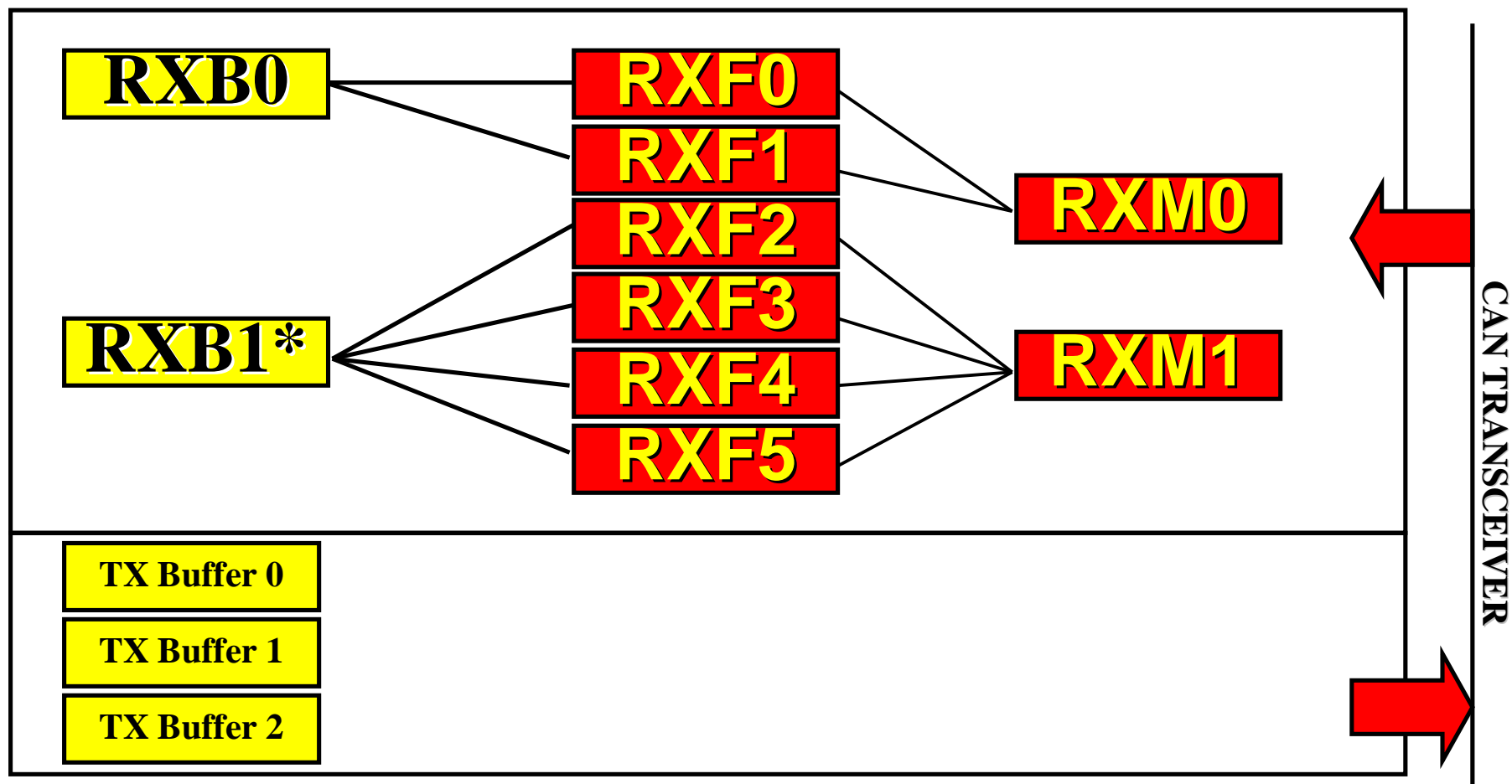
- | **Enhanced Controlled Area Network (ECAN)**
  - PIC18Fxx80
- | **CAN 2.0B Active (11 и 29-битные идентификаторы)**
- | **Обеспечивает**
  - Полную совместимость с модулями, встроенными в старые PIC
  - Дополнительно
    - 3 независимых режима
    - Дополнительные буферы, фильтры и маски
  - Поддержка DeviceNet™, RTR, FIFO

# Режим 0

- | **Режим 0: «Совместимый»**
  - | Режим совместим с MCP2515, PIC18C658 и PIC18F458
  - | Предназначен для простого перехода на новые контроллеры
- | **Ресурсы режима 0**
  - | **3 отдельных передающих буфера**
  - | **2 отдельных приемных буфера**
  - | **Буфер для сборки сообщений**
  - | **6 фильтров**
  - | **2 маски**



# Режим 0

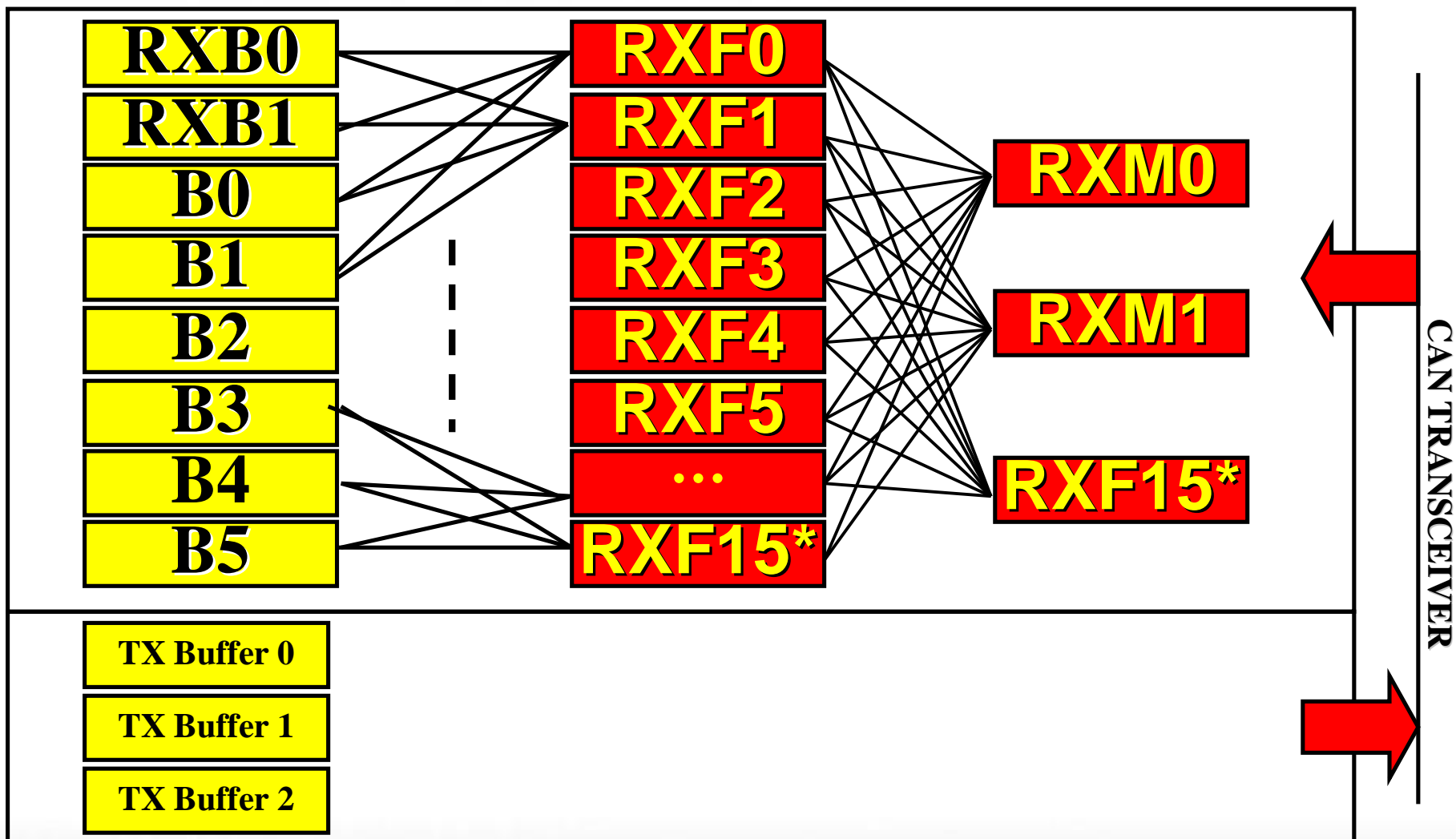


**\*При переполнении RXB0 можно писать в RXB1**

## Режим 1: «Расширенный совместимый»

- | Похож на режим 0 с дополнительными ресурсами
  - | Ресурсы режима 1
    - | 3 отдельных передающих буфера
    - | 2 отдельных приемных буфера
    - | 6 программируемых на прием или передачу
    - | Буфер для сборки сообщений
    - | 15 или 16 фильтров\*
    - | 2 или 3 маски\*
  - | Обработка RTR
  - | Программируемая фильтрация стандартных сообщений для реализации DeviceNet™
- \* Фильтр 15 можно использовать или как фильтр, или как маску

# Режим 1



\* RXF15 может быть фильтром или маской

# Режим 2

## Режим 2: «Расширенный режим с FIFO»

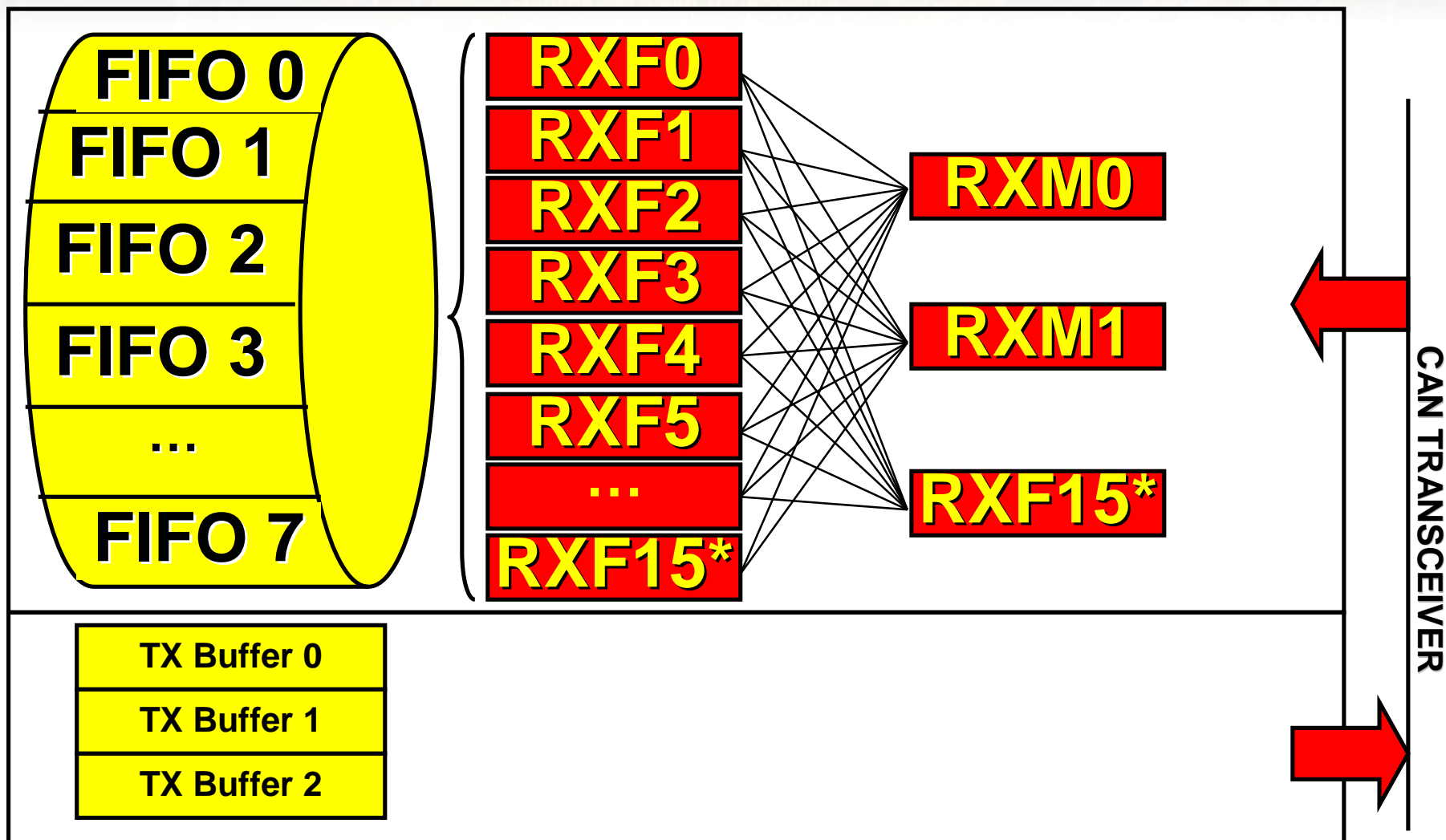
- | Входные буферы можно объединять в буфер FIFO
- | Фильтры и маски ассоциируются с буфером FIFO, а не с отдельным буфером
- | Ресурсы режима 2
  - | **3 передающих буфера**
  - | **2 приемных буфера**
  - | **6 программируемых на прием или передачу**
  - | **Буфер для сборки сообщений**
  - | **15 или 16 фильтров**
  - | **2 или 3 маски**

Обработка RTR

Программируемая фильтрация входных сообщений для реализации DeviceNet™

**\* Фильтр 15 можно использовать или как фильтр, или как маску**

# Режим 2



\* RXF15 может быть фильтром или маской





# Поддержка

- | **Вся информация**
- | **[www.microchip.com/can](http://www.microchip.com/can)**



# Ethernet

**увидите и освоите на  
специальном практическом  
занятии**