

# BSP и встраивание

## Методология встраивания ОС QNX

Трофимов Александр  
SWD Software Ltd.

Параметр \ CPU	Pentium II 233 МГц	Pentium II 350 МГц	PowerPC MTX604 300 МГц
Время реакции на прерывание, мкс	2.08	1.20	0.96
Время постановки потока на выполнение, мкс	5.46	4.18	4.65
Время отработки вызова ядра (sched_yield), мкс	1.62	1.30	1.85
Переключение контекста между потоками одного процесса, мкс	2.33	1.87	1.9

# QNX – ОС жесткого реального времени

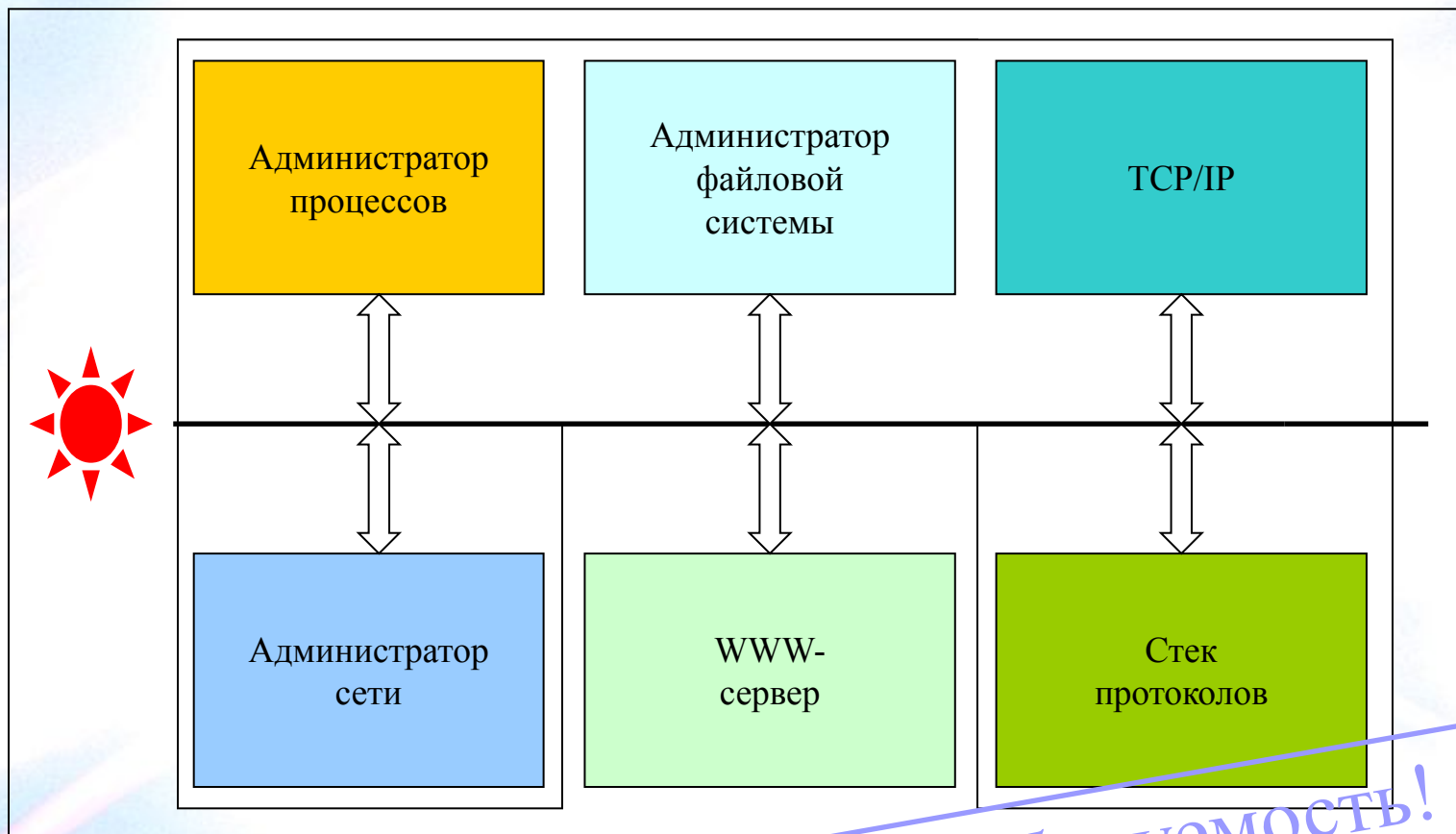


## QNX - ОС жесткого реального времени



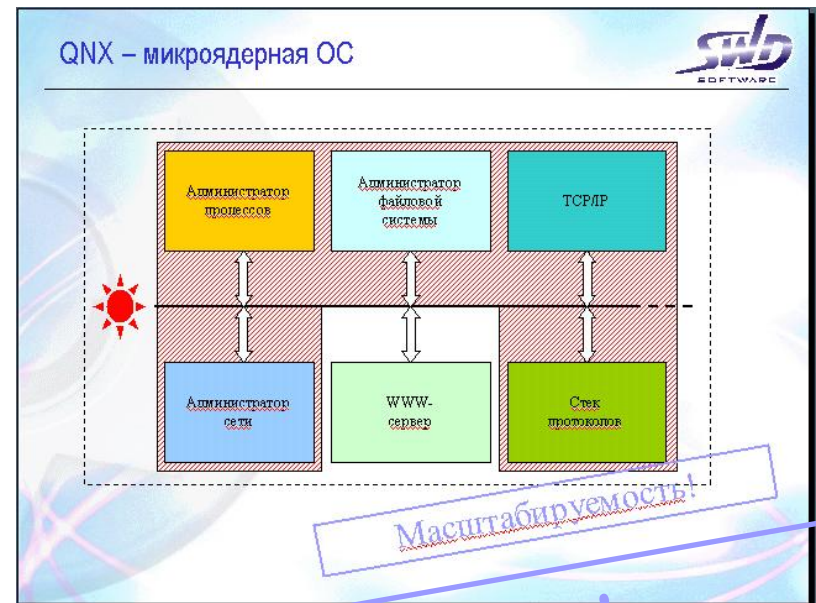
Параметр \ CPU	Pentium II 233 МГц	Pentium II 350 МГц	PowerPC MTX604 300 МГц
Время реакции на прерывание, <u>мкс</u>	2.08	1.20	0.96
Время постановки потока на выполнение, <u>мкс</u>	5.46	4.18	4.65
Время отработки вызова ядра ( <u>sched_yield</u> ), <u>мкс</u>	1.62	1.30	1.85
Переключение контекста между потоками одного процесса, <u>мкс</u>	2.33	1.87	1.9

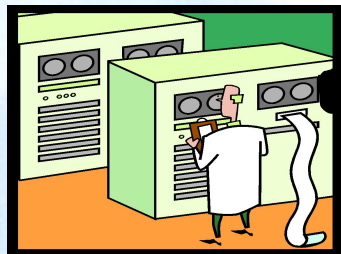




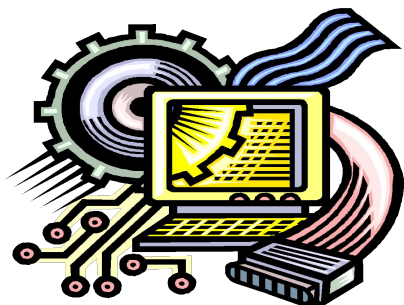
Масштабируемость!

- Гибко масштабируется
- Сохраняет ключевые свойства даже в минимальных конфигурациях
- QNX – многоплатформенная ОС
- QNX – ОС жесткого реального времени





✓ Особые требования к габаритам и энергопотреблению



✓ Особые требования к конструктивному исполнению

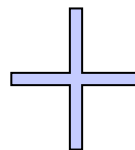
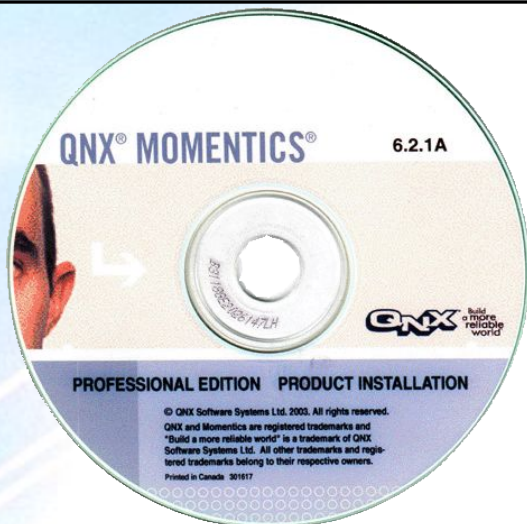


✓ Особые требования к вибро-, термо-, и удароустойчивости

- Базируются на различных процессорных платформах (PowerPC, ARM, x86 и т.д.)
- Обладают ограниченными ресурсами
- Обычно оснащены flash-носителями данных (DiskOnChip, Compact Flash, Onboard flash chip)



# Пакеты поддержки плат (BSP)



**BSP**

Пакет поддержки плат – продукт, содержащий все необходимые компоненты для установки и работы QNX на определенной процессорной плате



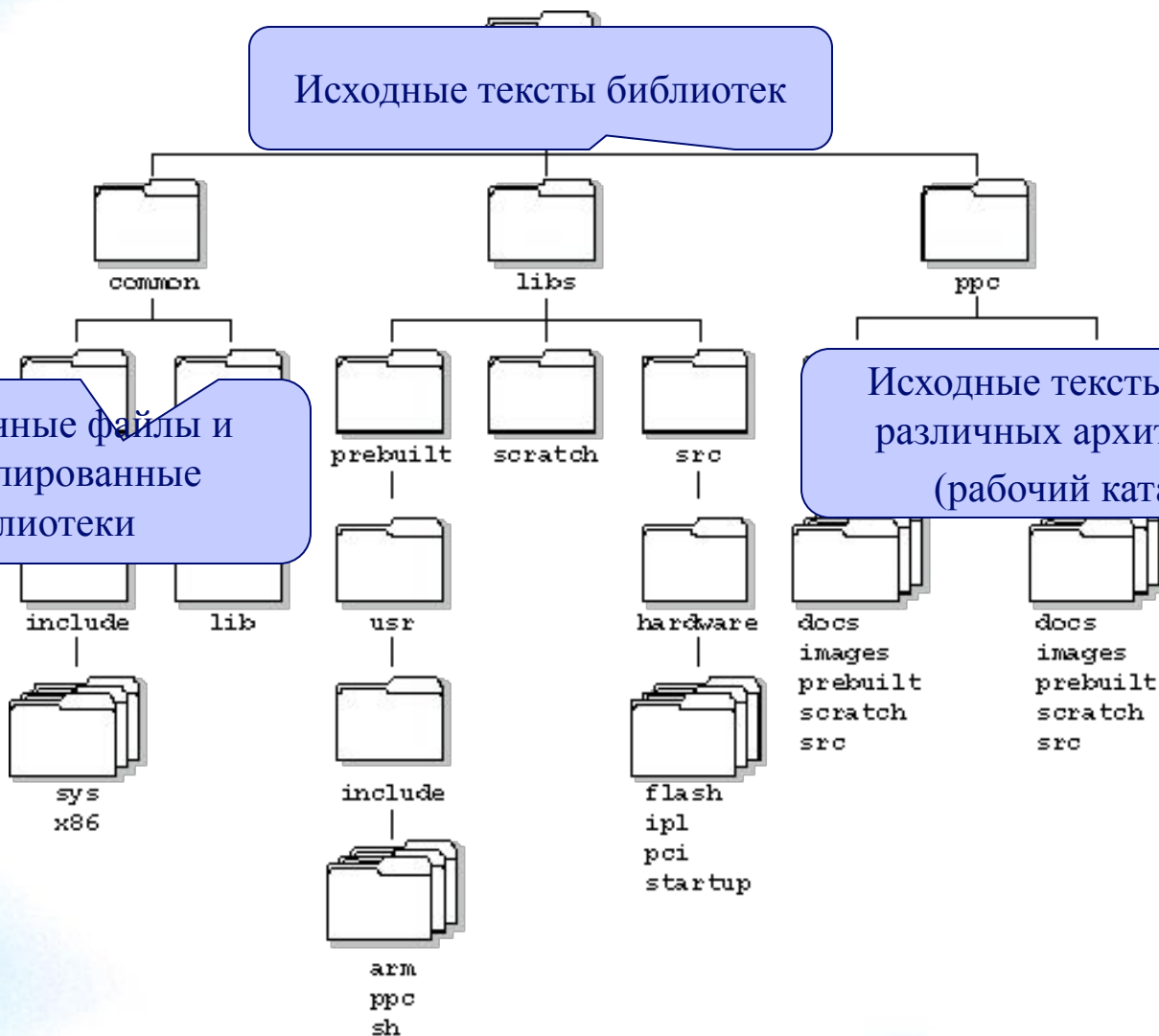
- ✓ Модуль первоначального загрузчика IPL
- ✓ Startup модуль
- ✓ Модуль поддержки файловой системы на флеш-носителе
- ✓ PCI сервер

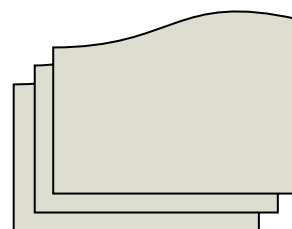


## Исходные тексты библиотек

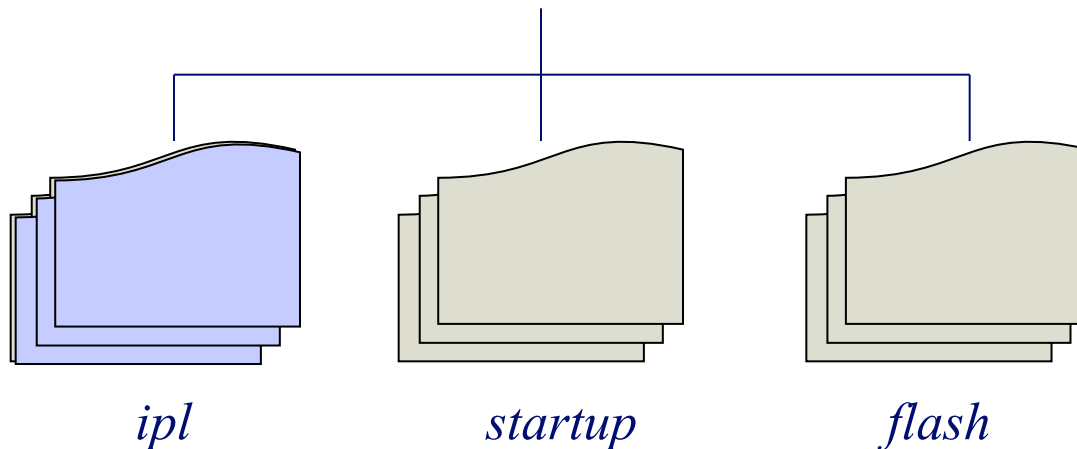
Заголовочные файлы и  
скомпилированные  
библиотеки

Исходные тексты модулей для  
различных архитектур и плат  
(рабочий каталог BSP)





*bsp working dir/src/hardware*



IPL размещается на векторе сброса процессора, при передаче ему управления производит следующие действия:

- ✓ Минимальная инициализация оборудования;
- ✓ Загрузка образа ОС из носителя данных в ОЗУ;
- ✓ Определение адреса ОЗУ, по которому размещается загруженный образ ОС;
- ✓ Копирование startup-модуля в ОЗУ;
- ✓ Передача управления startup-модулю;

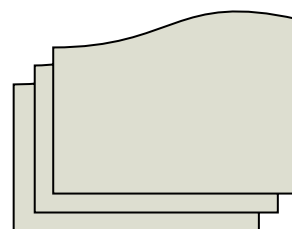
## Ассемблерный код

```
main() {  
  
    image_download_hw()  
  
    image_scan()  
  
    image_setup()  
  
    image_start()  
  
}
```

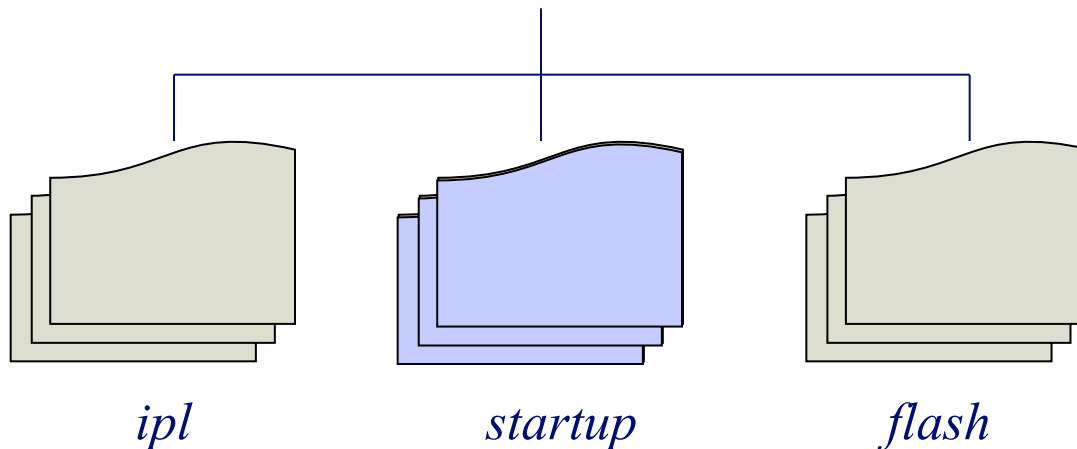
- init\_myboard.s

- main.c

Большая часть кода IPL  
– это вызовы  
библиотечных функций  
IPL library



*bsp working dir/src/hardware*



Startup-модуль – это самая первая программа в загружаемом образе Neutrino. Она получает управление от IPL и выполняет следующие действия:

- ✓ “Доинициализация” оборудования;
- ✓ Сбор информации о системе и заполнение специальной структуры данных, так называемой system page;
- ✓ Инициализация callout-ов микроядра;
- ✓ Загрузка следующей программы из образа и передача ей управления;

Global variables

main()

{

Call add\_callout\_array()

Argument parsing (Call  
handle\_common\_option())

Call init\_raminfo()

Remove ram used by modules in the image

if (virtual) Call init\_mmu() to initialize the MMU

Call init\_intrinfo()

Call init\_qtime()

Call init\_cacheattr()

Call init\_cpuinfo()

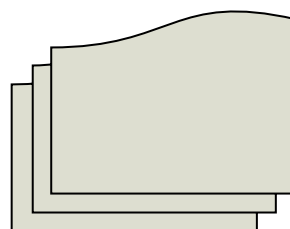
Set hardware machine name

Call init\_system\_private()

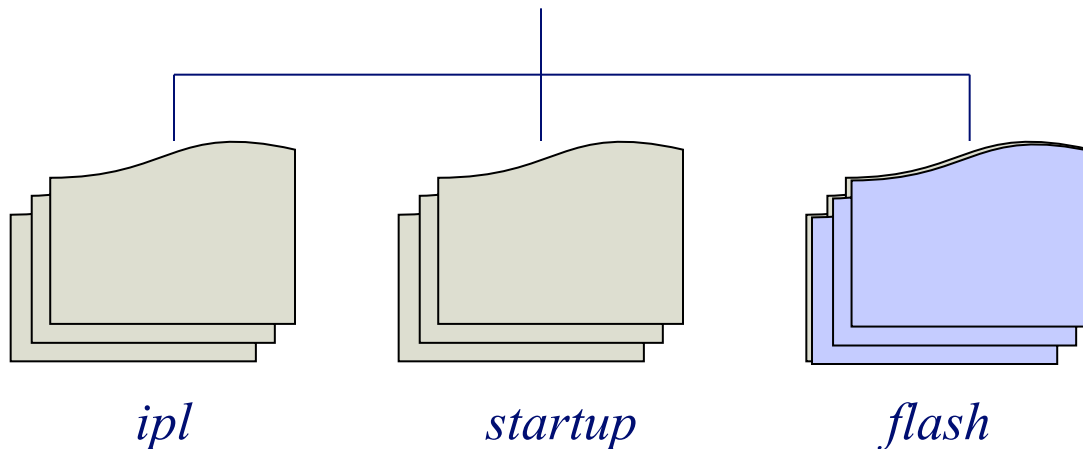
Call print\_syspage() to print debugging output

}

Как и в IPL большая часть кода – вызовы библиотечных функций startup-library для опроса конфигурации оборудования, инициализации system page, загрузки в ОЗУ следующего процесса из образа ОС, переключения процессора в защищенный режим

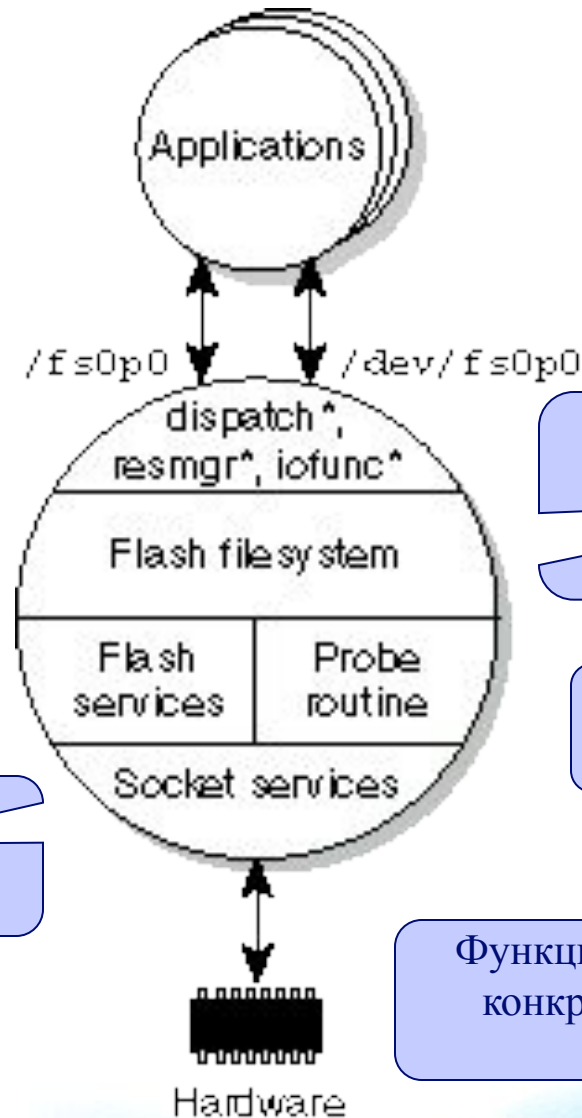


*bsp working dir/src/hardware*





# Драйвер встроенной файловой системы



Содержит основную логику flash filesystem. Доступ к оборудованию осуществляется при помощи нижележащих компонентов. Находится в библиотеке libfs-flash.a

Инициализация и адресация flash устройств

Как менеджер ресурсов драйвер должен обрабатывать все стандартные сообщения

Определение размера flash массива

Функции записи и стирания для конкретного flash устройства (MTD)

```
int main(int argc, char **argv) {
int error;
static f3s_service_t service[]= {
.....
};
static f3s_flash_t flash[]= {
.....
}
/* init f3s */
F3s_init(argc, argv, flash);
/* start f3s */
error=f3s_start(service, flash);
return error;
}
```

Данный код реализуется  
исходя из особенностей  
доступа и адресации  
flash устройства

Для ряда flash  
чипов данный код  
реализован в MTD  
(libmtd-flash.a)  
библиотеке из  
состава BSP

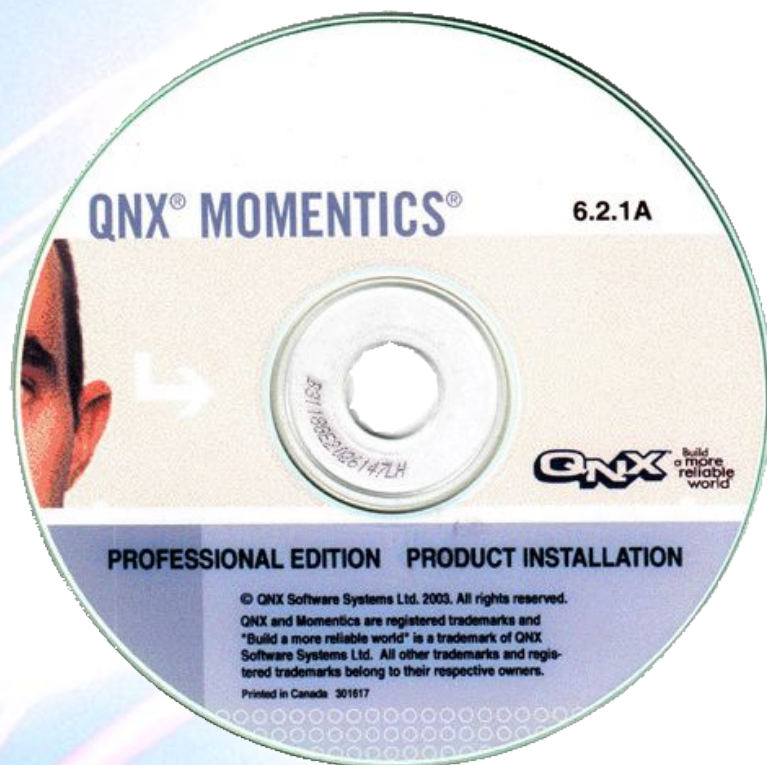
В библиотеке libmtd-flash.a реализованы функции MTD уровня для некоторых flash чипов различных производителей:

- AMD
- Intel
- Sharp
- Fujitsu
- Псевдоустройства (например, оперативная память)



В общей  
сложности  
более 120  
устройств

## Пакеты поддержки плат



- ✓ В состав комплекта разработчика QNX Momentics 6.2.1 PE входит более **17 BSP** для более чем **25** процессорных плат;
- ✓ Список доступных BSP постоянно расширяется;

# Fastwel CPU686E - Стартовый комплект

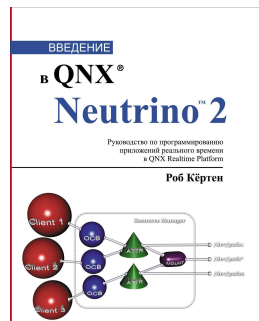


Процессоры  
предустановленной  
средой исполнения QNX,  
разъемы, кабели

**Бесплатно!**



Комплект разработчика  
QNX Momentics PE/SE



Литература и инстру  
на русском языке,  
документация,  
учебные видеоролики

**Бесплатно!**

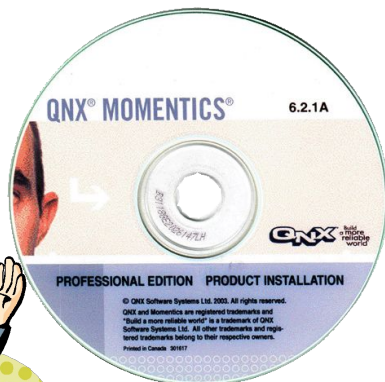
Стартовый комплект =  
готовый набор  
инструментов для  
разработки ПО



# BSP и встраивание Практические приемы

Трофимов Александр  
SWD Software Ltd.

## IBM PPC 405GP Walnut



В состав QNX Momentics  
6.2.1 входит BSP для IBM  
PPC 405GP Walnut

BSP для IBM PPC 405GP (Walnut) входит в состав QNX Momentics 6.2.1 и состоит из следующих компонентов:

- ✓ IPL модуль с исходными кодами;
- ✓ Startup модуль с исходными кодами;
- ✓ PCI сервер с исходными кодами;
- ✓ Сценарии загрузочных образов системы;
- ✓ Исчерпывающая документация;



Установив QNX вместе с BSP для Walnut на инструментальной машине, можно выбирать различные варианты дальнейшей работы с целевой системой:

- ✓ **Загрузка по протоколу bootp;**
- ✓ **Загрузка при помощи IPL;**

Из компонентов BSP собирается загрузочный образ целевой системы.

На инструментальной системе настраиваются bootp и tftp сервер для передачи образа целевой системе.

# Пример №1 – IBM PPC 405GP Walnut

Инструментальная система



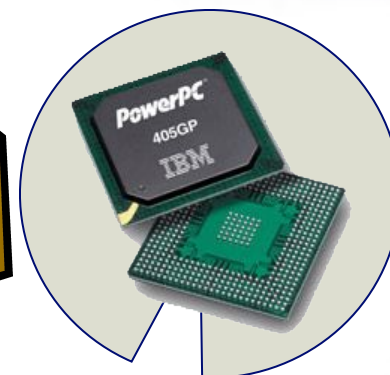
Ethernet



Соединение

Нуль-модемное соединение  
COM портов

Целевая система



- ✓ Через последовательное соединение конфигурируется ROM Monitor целевой системы;
- ✓ Передается образ по сети;

# Пример №1 – IBM PPC 405GP Walnut

```
ttyp0: qtalk

405GP 1.13 ROM Monitor (4/7/00)

----- System Info -----
Processor           = 405GP,   PUR: 40110082
Processor speed    = 200 MHz
PLB speed          = 100 MHz
OPB speed         = 50 MHz
Ext Bus speed     = 50 MHz
PCI Bus speed     = 33 MHz (Sync)
Amount of SDRAM   = 32 MBytes
Internal PCI arbiter enabled

-----

--- Device Configuration ---
Power-On Test Devices:
000 Disabled System Memory [RAM]
001 Enabled Ethernet [ENET]
004 Enabled Serial Port 2 [S2]

-----

Boot Sources:
001 Enabled Ethernet [ENET]
      local=192.168.3.34 remote=192.168.3.32 hwaddr=0004ace30f97
004 Disabled Serial Port 2 [S2]
      local=8.1.1.2 remote=255.255.255.255 hwaddr=ffffffffffff
005 Disabled Serial Port 1 [S1]

-----

Debugger: Disabled

-----

1 - Enable/disable tests
2 - Enable/disable boot devices
3 - Change IP addresses
4 - Ping test
5 - Toggle ROM monitor debugger
6 - Toggle automatic menu
7 - Display configuration
8 - Save changes to configuration
9 - Set baud rate for s1 boot
A - Enable/disable I cache (Disabled)
B - Enable/disable D cache (Enabled)
0 - Exit menu and continue

->_
```

# Пример №1 – IBM PPC 405GP Walnut

```
ttyp0: qtalk
Ext Bus speed      = 50 MHz
PCI Bus speed      = 33 MHz (Sync)
Amount of SDRAM    = 32 MBytes
Internal PCI arbiter enabled
-----

--- Device Configuration ---
Power-On Test Devices:
000 Disabled System Memory [RAM]
001 Enabled Ethernet [ENET]
004 Enabled Serial Port 2 [S2]
-----

Boot Sources:
001 Enabled Ethernet [ENET]
      local=192.168.3.34 remote=192.168.3.32 hwaddr=0004ace30f97
004 Disabled Serial Port 2 [S2]
      local=8.1.1.2 remote=255.255.255.255 hwaddr=ffffffffffff
005 Disabled Serial Port 1 [S1]
-----

Debugger: Disabled
-----

1 - Enable/disable tests
2 - Enable/disable boot devices
3 - Change IP addresses
4 - Ping test
5 - Toggle ROM monitor debugger
6 - Toggle automatic menu
7 - Display configuration
8 - Save changes to configuration
9 - Set baud rate for s1 boot
A - Enable/disable I cache (Disabled)
B - Enable/disable D cache (Enabled)
0 - Exit menu and continue
->3
----- CHANGE IP ADDRESS -----
Device List:
001 Enabled Ethernet [ENET]
      local=192.168.3.34 remote=192.168.3.32 hwaddr=0004ace30f97
004 Disabled Serial Port 2 [S2]
      local=8.1.1.2 remote=255.255.255.255 hwaddr=ffffffffffff
-----
select device to change ->_
```

# Пример №1 – IBM PPC 405GP Walnut

```
ttyp0: qtalk
local=192.168.3.34 remote=192.168.3.32 hwaddr=0004ace30f97
004 Disabled Serial Port 2 [S2]
local=8.1.1.2 remote=255.255.255.255 hwaddr=ffffffffffff
-----
select device to change ->00
1 - Change local address
2 - Change remote address
0 - Return to main menu
->0
1 - Enable/disable tests
2 - Enable/disable boot devices
3 - Change IP addresses
4 - Ping test
5 - Toggle ROM monitor debugger
6 - Toggle automatic menu
7 - Display configuration
8 - Save changes to configuration
9 - Set baud rate for s1 boot
A - Enable/disable I cache (Disabled)
B - Enable/disable D cache (Enabled)
0 - Exit menu and continue
->4
--- PING TEST ---
Device List:
001 Enabled Ethernet [ENET]
local=192.168.3.34 remote=192.168.3.32 hwaddr=0004ace30f97
004 Disabled Serial Port 2 [S2]
local=8.1.1.2 remote=255.255.255.255 hwaddr=ffffffffffff
-----
select device to ping ->1
ENET Speed is 100 Mbs...
FULL duplex connection
Using [ENET] to ping. press any key to stop.
PING 192.168.3.32 56 data bytes
68 bytes from 192.168.3.32: icmp_seq=0 ttl=255 time=6 ms
68 bytes from 192.168.3.32: icmp_seq=1 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=2 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=3 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=4 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=5 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=6 ttl=255 time=3 ms
```

# Пример №1 – IBM PPC 405GP Walnut

```
ttyp0: qtalk
68 bytes from 192.168.3.32: icmp_seq=2 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=3 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=4 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=5 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=6 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=7 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=8 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=9 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=10 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=11 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=12 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=13 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=14 ttl=255 time=3 ms
-- 192.168.3.32 ping statistics --
15 packets transmitted, 15 packets received, 0% packet loss
 1 - Enable/disable tests
 2 - Enable/disable boot devices
 3 - Change IP addresses
 4 - Ping test
 5 - Toggle ROM monitor debugger
 6 - Toggle automatic menu
 7 - Display configuration
 8 - Save changes to configuration
 9 - Set baud rate for s1 boot
 A - Enable/disable I cache (Disabled)
 B - Enable/disable D cache (Enabled)
 0 - Exit menu and continue
->0

EMAC Ethernet Test OK.
Serial Port 2 [S2] test complete - TEST PASSED
Booting from [ENET] Ethernet
MAL error occured... ISR = c0100010 UIC = = 2a0000 MAL_DEF = 2e0000 MAL_ERR=
260000
ENET Speed is 100 Mbs...
FULL duplex connection
Sending bootp request ...

Loading file "/tftpboot/walnut.ifs" ...
Sending tftp boot request ...
_block 358
```

# Пример №1 – IBM PPC 405GP Walnut

```
ttyp0: qtalk
68 bytes from 192.168.3.32: icmp_seq=6 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=7 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=8 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=9 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=10 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=11 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=12 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=13 ttl=255 time=3 ms
68 bytes from 192.168.3.32: icmp_seq=14 ttl=255 time=3 ms
--- 192.168.3.32 ping statistics ---
15 packets transmitted, 15 packets received, 0% packet loss
 1 - Enable/disable tests
 2 - Enable/disable boot devices
 3 - Change IP addresses
 4 - Ping test
 5 - Toggle ROM monitor debugger
 6 - Toggle automatic menu
 7 - Display configuration
 8 - Save changes to configuration
 9 - Set baud rate for s1 boot
 A - Enable/disable I cache (Disabled)
 B - Enable/disable D cache (Enabled)
 0 - Exit menu and continue
->0

EMAC Ethernet Test OK.
Serial Port 2 [S2] test complete - TEST PASSED
Booting from [ENET] Ethernet
MAL error occured... ISR = c0100010 UIC = = 2a0000 MAL_DEF = 2e0000 MAL_ERR=
260000
ENET Speed is 100 Mbs...
FULL duplex connection
Sending bootp request ...

Loading file "/tftpboot/walnut.ifs" ...
Sending tftp boot request ...
Transfer Complete ...
Loaded successfully ...
Entry point at 0x3177c ...
Welcome to Neutrino on the PPC 405 board
# _
```

Установив QNX вместе с BSP для Walnut на инструментальной машине, можно выбирать различные варианты дальнейшей работы с целевой системой:

- ✓ Загрузка по протоколу bootp;
- ✓ Загрузка при помощи IPL;



Walnut оснащен 29f040 совместимым flash массивом емкостью 512К. Изначально в нем прошит код ROM Monitor'a. При помощи СППЗУ программатора в массив можно прошить код для загрузки QNX несколькими способами:

- ✓ В массив прошивается только код IPL, сконфигурированный таким образом, чтобы загружать образ по последовательному соединению по протоколу sendnto;
- ✓ В массив прошивается как IPL, так и загрузочный образ системы. Размер массива накладывает ограничение на размер загружаемого образа в 504К;

# Пример №1 – IBM PPC 405GP Walnut

QNX System Builder - QNX Momentics IDE

File Edit Navigate Search Project Run Window Help

Navigator

walnut

- Images
- Overrides
- Reductions
- .project
- .sysbldr\_meta
- project.bld

Item Browser

Item Attributes

walnut.build

- Binaries
- Shared L
- Symbolic

Image Path:

Local Path:

Images Filesystem Boot Script Source

Serial Terminal

Device: /dev/ser1 Settings Send BREAK Clear Terminal Send File...

Send File

Select the protocol and file you wish to send

Protocol: sendnto

Filename: walnut.build.ifs Browse...

Applications

- Welcome
- Help
- Installer
- File Manager
- Voyager
- Editor
- Terminal
- Media Player
- Utilities
- Configure
- Network
- Mouse

Launch QNX System

4AM

Интегрированная среда разработки – QNX IDE позволяет строить загрузочные образы и передавать их на целевую систему

## Fastwel CPU686E



- ✓ Специалистами SWD Software Ltd. был разработан пакет поддержки Fastwel CPU686E для QNX Neutrino;
- ✓ SE пользователям пакет предоставляется в виде исполняемых модулей, пользователям PE пакет предоставляется вместе с исходными текстами;

Fastwel CPU686E - Стартовый комплект

QNX 2003  
РОССИЯ  
МЕЖДУНАРОДНАЯ КОМПЕТИЦИЯ

 Процессоры  
предустановленной  
средой исполнения QNX,  
разъемы, кабели **Бесплатно!**

 Комплект разработчика  
QNX Momentics PE/SE

 Литература и инстру  
на русском языке,  
документация,  
учебные видеоролики **Бесплатно!**

Стартовый комплект =  
готовый набор  
инструментов для  
разработки ПО



При разработке пакета поддержки Fastwel CPU686E для QNX Neutrino были проанализированы следующие характеристики целевой системы:



- ✓ Fastwel CPU686E построена на платформе x86;
- ✓ Fastwel CPU686E оснащена BIOS;
- ✓ Fastwel CPU686E оснащена 8 Мб NAND Flash массивом;

## Результаты анализа

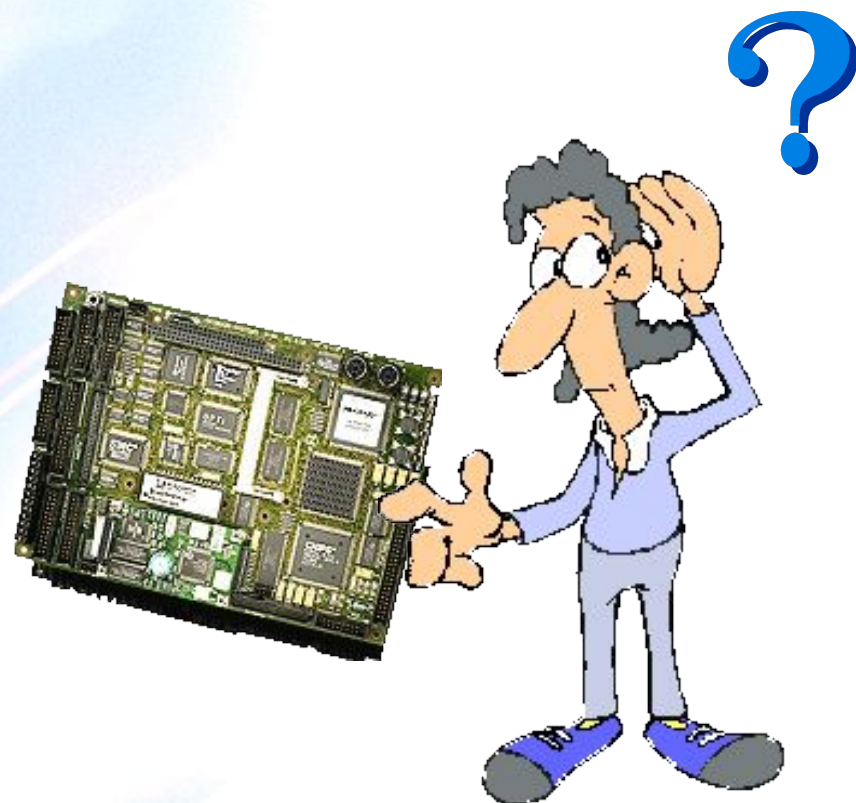


- ✓ За основу пакета можно взять наиболее «близкий» к Fastwel BSP для платы Elan SC400;
- ✓ В качестве PCI сервера можно использовать стандартный pci-bios;
- ✓ Startup модуль из BSP для Elan SC400 полностью подходит для Fastwel CPU686E;
- ✓ Доступ к NAND Flash массиву осуществляется через порты ввода-вывода, а не через окно памяти;
- ✓ MTD функции для NAND Flash массива не реализованы в MTD библиотеке BSP;

## Что нужно сделать



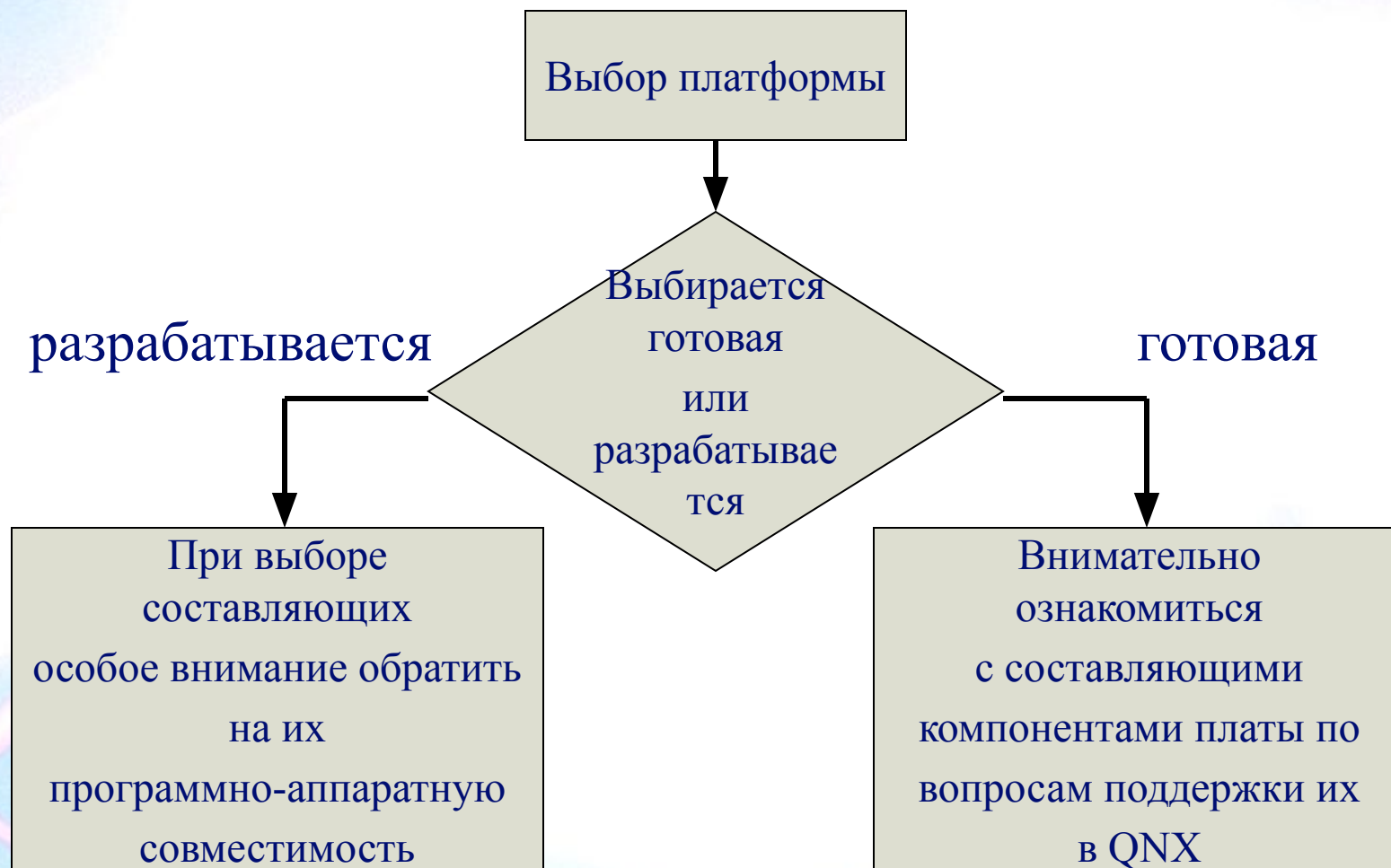
- ✓ Добавить в MTD библиотеку функции MTD для работы с NAND Flash;
- ✓ Поскольку доступ к Flash осуществляется через порты ввода-вывода, необходимо переписать функции flash services;
- ✓ В IPL реализовать способ считывания образа QNX Neutrino в оперативную память с Flash массива;



- ✓ В общем случае программную и аппаратную часть системы создают разные группы разработчиков;
- ✓ При выборе аппаратной части необходимо обратить особое внимание на программно-аппаратную совместимость;



## Алгоритм выбора платы





Мастер-класс Игоря Мазанова, посвященный вопросам **«программно-аппаратной совместимости в ОС QNX»**, состоится 20 сентября 2003 в 13:30 в зале «CARMEL»

## Когда плата уже выбрана

- ✓ Проверить наличие готового BSP для данной платы;
- ✓ Если нет, сравнить выбранную плату с теми, для которых существует готовый BSP, и найти среди них наиболее «близкую»;
- ✓ Взять BSP для «близкой» платы за основу проекта;
- ✓ Проанализировать различия между данной и «близкой» платой;
- ✓ Доработка модулей «близкого» BSP, требующих модификации;

## Анализ различий

- ✓ Алгоритм загрузки;
- ✓ Совместимость PCI контроллера;
- ✓ Наличие onboard flash накопителя;
- ✓ Поддержка данного flash накопителя библиотекой MTD;
- ✓ Метод доступа к onboard flash;
- ✓ Оборудование, которое необходимо инициализировать на различных этапах загрузки;

# Спасибо!



SWD Software Ltd.  
Официальный дистрибьютор  
QNX

196135, Санкт-Петербург,  
пр. Юрия Гагарина 23  
тел.: (812) 102-0833  
тел.: (812) 373-0260  
факс: (812) 373-0497  
web: <http://www.swd.ru/>  
e-mail: [qnx@swd.ru](mailto:qnx@swd.ru)