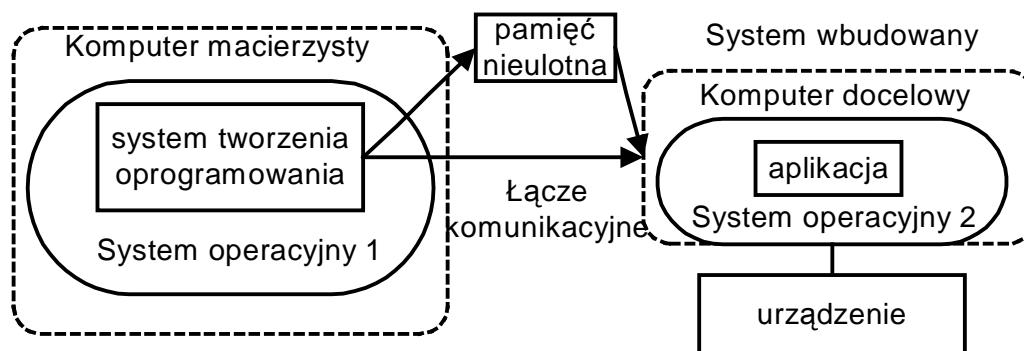


Tworzenie aplikacji wbudowanych

1. Tworzenie oprogramowania systemów wbudowanych

System wbudowany (ang. *embedded system*)

System wbudowany jest to system komputerowy będący częścią większego systemu i wykonujący istotną część jego funkcji. Przykładem może być komputer pokładowy samolotu lub system sterujący szybką koleją miejską.



Rys. 1-1 System skrótnego rozwoju oprogramowania

Narzędziem do tworzenia takich aplikacji jest Momentics Development Suite firmy QNX Software.

Aplikacje mogą być tworzone w środowiskach systemów QNX6 Neutrino, Windows (NT, 2000, XP), Linux, Solaris SPARC.

Środowisko to zawiera:

- Zintegrowane środowisko rozwoju oprogramowania IDE (*ang. Integrated Development Environment*).
- Biblioteki ANSI C, POSIX, DINKUM C++, GNU C++.
- Dokumentację.
- Narzędzia BSP (*ang. Board Support Packages*) służące do dostosowania systemu QNX6 Neutrino do różnorodnych środowisk sprzętowych.
- Narzędzia DDK (*ang. Driver Development Kits*) wspomagające tworzenie sterowników urządzeń zewnętrznych (grafiki, audio, sieciowe, USB).
- Narzędzia TDK (*ang. Technology Development Kits*) które ułatwiają tworzenie wbudowanych systemów plików, systemów wieloprocessorowych SMP, obsługi multimediiów i grafiki 3D.

System ten może być zainstalowany w komputerach z procesorami ARM, MIPS, PowerPC, SH-4, StrongARM (w tym Xscale) i x86.

2. Sprzęt systemów wbudowanych

Wymagania na komputery przeznaczone do sterowania i zastosowań wbudowanych.

1. Wymagana jest odporność na pracę w trudnych warunkach otoczenia (wibracje, zapylenie, wilgoć), dopuszczalny jest szeroki zakres temperatur otoczenia.
2. Przeznaczone są do pracy ciągłej - brak jest elementów ruchomych (dyski obrotowe, wentylatory, napędy dyskietek), wymagana jest trwałość, łatwość serwisowania.
3. Oprogramowanie umieszczone jest w pamięci nieulotnej – ROM, flash, EPROM lub podobnej.

Stosowane jest wsparcie sprzętowe dla osiągnięcia niezawodnej pracy – budzik (*ang. watchdog*), pamięci ECC, magistrala z kontrolą parzystości, poszerzona diagnostyka.

Rozwiązania:

- Standard PC104
- Standard Compact PCI
- Magistrala VME

3. Standard PC104

Standard PC104 ma na celu dostosowanie komputerów do warunków przemysłowych i systemów wbudowanych. Definiuje różnice mechaniczne, elektryczne i logiczne względem standardu ISA i EISA.

Standard utrzymywany przez: PC/104 Consortium
<http://www.pc104.org>

Najważniejsze różnice względem magistrali ISA:

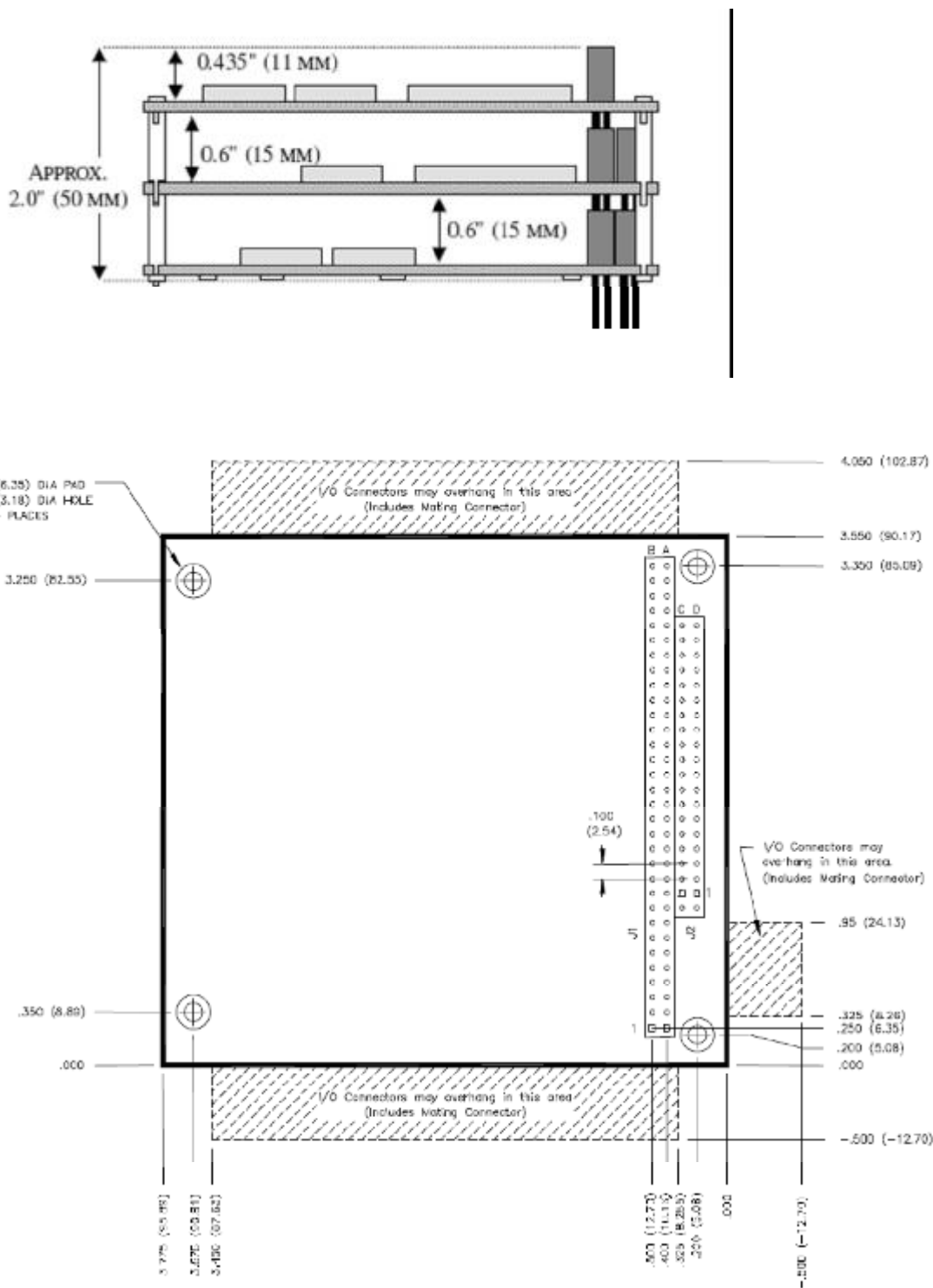
1. Wymiary 90 x 96 mm.
2. Eliminacja potrzeby podstawy (*ang. backplane*) i obudowy (konstrukcja jest samonośna i kanapkowa).
3. Redukcja liczby elementów i zapotrzebowania na zasilanie (typowo 1-2 W na moduł) Wymagana moc wysterowania pojedynczego sygnału magistrali wynosi 4mA.

Standard dopuszcza 2 rodzaje modułów – 8 bitowe i 16 bitowe.

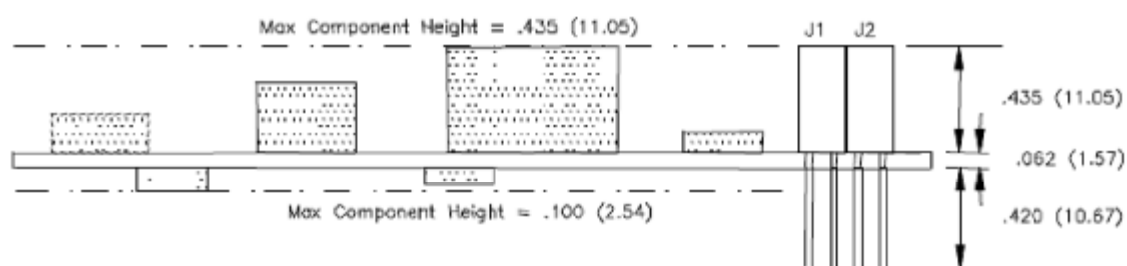
Standard wprowadza możliwość dzielenia jednego przerwania przez wiele urządzeń.

1.1. Standard mechaniczny

Standaryzacji podlega konstrukcja i wymiary płytki.



Rzut poziomy - płytka 16 bitowa o wymiarach 90 x 96 mm.



Rzut pionowy - płytki 16 bitowa

Normalizacji podlegają też własności łączówek

1. Wymiary
2. Materiały z którego są wykonane
3. Jakość kontaktów
4. Własności elektryczne

1.2. Standard logiczny

Sygnaly magistrali:

1. Adresy i dane
2. Kontrola cyklu
3. Kontrola magistrali
4. Przerwania
5. DMA

Adresy, dane,

BALE	Bus Address Latch Enable	Gdy linia jest aktywna sygnaly SA<19:0>, LA<23:17>, AEN i SBHE są ważne.
SA<19:0>	Address	20 niższych linii adresowych (pierwsze 1MB)
LA<23:17>	Latched address	Wyższe linie adresowe (całe 16 MB)
IOCHRDY	Input Output Channel Ready	Linia aktywna gdy urz. we/wy żąda dodatkowych cykli oczekiwania
AEN	Address Enable	Poziom wysoki gdy DMA przejął kontrolę nad magistralą
SD<15:0>	Data	Linie danych
SBHE#	System Bus High Enable	Sygnal wystawiany przez procesor lub DMA i wskazuje że dane na liniach SD<8-15> są ważne

Kontrola cyklu

IOR#	Input Output Read	Linia aktywowana przez procesor lub DMA gdy ma nastąpić odczyt z urz. we/wy
IOW#	Input Output Write	Linia aktywowana przez procesor lub DMA gdy ma nastąpić zapis do urz. we/wy
MEMR#	Memory Read	Linia aktywowana przez procesor lub DMA gdy potrzebny jest odczyt z pamięci
SMEMR#	System Memory Read	Linia aktywowana gdy potrzebny jest odczyt z pierwszego 1 MB pamięci
MEMW#	Memory Write	Linia aktywowana przez procesor lub DMA gdy potrzebny jest zapis do pamięci
SMEMW#	System Memory Write	Linia aktywowana gdy potrzebny jest zapis do pierwszego 1 MB pamięci
MEMCS16#	Memory Chip Select 16	Dostęp do pamięci w trybie 16 bitowym
IOCS16#	IO Chip Select 16	Dostęp do urządzenia we/wy w trybie 16 bitowym

Kontrola magistrali

RESET	Reset	Przekazuje sygnał resetu
BCLK	Bus Clock	Zegar magistrali od 6 do 8 MHz
OSC	Oscilator	Sygnał zegara systemowego 14.318180 MHz
IOCHK#	IO Chanel Check	Sygnał zgłaszania błędów układów rozszerzających
REFRESH#	Refresh	Sygnał odświeżania pamięci
MASTER16#	Master16	Linia aktywowana przez układ typu Master znajdujący się na karcie rozszerzającej gdy przejmuje on kontrolę nad magistralą.

Sterowanie transmisją DMA

DRQ1 - 7	DMA Request	Linia żądania przydziału kanału DMA
DACK1 - 7	DMA Acknowledge	Potwierdzenie przyjęcia żądania DMA
TC	Terminal Count	Sygnał generowany przez kontroler DMA wskazujący że wszystkie dane zostały przesłane

Przerwania

IRQ3 - 7	Interrupt Request	Linie zgłoszenia przerwań – 8bit
IRQ10, 11, 12, 14, 14	Interrupt Request	Linie zgłoszenia przerwań – 16 bit

J2/P2			J1/P1		
Pin	Row D	Row C	Pin	Row A	Row B
0	GND	GND	1	IOCHK*	GND
1	MEMCS16*	SBHE*	2	SD7	RESET
2	IOCS16*	LA23	3	SD6	+5V
3	IRQ10	LA22	4	SD5	IRQ9
4	IRQ11	LA21	5	SD4	-5V
5	IRQ12	LA20	6	SD3	DRQ2
6	IRQ15	LA19	7	SD2	-12V
7	IRQ14	LA18	8	SD1	SRDY*
8	DACK0*	LA17	9	SD0	+12V
9	DRQ0	MEMR*	10	IOCHRDY	KEY
10	DACK5*	MEMW*	11	AEN	SMEMW*
11	DRQ5	SD8	12	SA19	SMEMR*
12	DACK6*	SD9	13	SA18	IOW*
13	DRQ6	SD10	14	SA17	IOR*
14	DACK7*	SD11	15	SA16	DACK3*
15	DRQ7	SD12	16	SA15	DRQ3
16	+5V	SD13	17	SA14	DACK1*
17	MASTER*	SD14	18	SA13	DRQ1
18	GND	SD15	19	SA12	REFRESH*
19	GND	KEY	20	SA11	BCLK
			21	SA10	IRQ7
			22	SA9	IRQ6
			23	SA8	IRQ5
			24	SA7	IRQ4
			25	SA6	IRQ3
			26	SA5	DACK2*
			27	SA4	TC
			28	SA3	BALE
			29	SA2	+5V
			30	SA1	OSC
			31	SA0	GND
			32	GND	GND

Rozmieszczenie sygnałów na magistrali

4. Budowa komputera PM-1041

Komputer PM-1041 jest jednopłytkowym komputerem zbudowanym według standardu PC104.

Komputer udostępniony dzięki uprzejmości firmy:

QUANTUM Sp. z o.o. - Korporacja Transferu Technologii

ul. Skwierzyńska 21

53-521 Wrocław

<http://www.quantum.com.pl/>

4.1 Własności

Specyfikacja

CPU	486DX2-66 STPC
Pamięć	Jeden moduł 144-pin SO-DIMM do 32MB EDO RAM
Chipset	SGS Thomson STPC-Client
Szyna systemowa	Złącze PC104
Kontroler monitora (STPC)	VRAM : 512KB ~ 4MB, pamięć pobierana z systemowej Rozdzielczość : 1024 x 768 (64 kolory)
Urządzenia WE/WY	2 x port szeregowy (kompatybilny z 16550 UART) 1 x port równoległy (wsparcie SPP/EPP/ECP) IrDA port
Kontroler dysku IDE	Dla 44-pin taśmy, wspiera tryb PIO-4
Kontroler stacji dyskietek	Jeden port kontrolera dyskietek
Watchdog Timer	Programowalny Watchdog Timer
Dysk półprzewodnikowy	Gniazdo na DiskOnChip™
Klawiatura, myszka	Standardowe gniazdo PS2
Zapotrzebowanie mocy	5V, 1.4A
Temperatura pracy	0-55 stopni Celsjusza



4.2 Watchdog

Watchdog jest urządzeniem zabezpieczającym przed zatrzymaniem wykonywania programów które może nastąpić przy awarii sprzętu lub błędzie programowym.

Port	Operacja	Funkcja
0x443	Zapis	Ustawienie okresu odświeżania
0x443	Odczyt	Odświeżanie watchdoga
0x843	Odczyt	Zablokowanie watchdoga

Funkcje portów watchdoga komputera PM-1041

Aktywacja watchdoga

Aktywacja watchdoga następuje poprzez wpis interwału odświeżania pod adres 0x443.

Zawartość	Okres odśwież
0x1	1 sek
0x2	2 sek
...	...
0xFF	255 sek

Odświeżanie watchdoga

Odświeżanie watchdoga następuje poprzez odczyt z portu 0x443. Gdy w zaprogramowanym wcześniej okresie nie nastąpi odczyt – watchdog dokona resetu systemu.

Zatrzymywanie watchdoga

Zatrzymywanie watchdoga następuje poprzez odczyt spod adresu 0x843.

```

/*-----*/
/* Program obsługi watchdog-a */
/* Kompilacja: cc wdog.c -o wdog -T0 */
/*-----*/
#include <stdio.h>
#include <i86.h>
#define WDOG_ENABLE 0x443 // Uruchomienie watchdoga
#define WDOG_REFRESH 0x443 // Odświeżenie watchdoga
#define WDOG_DISABLE 0x843 // Wylaczenie watchdoga

void main(int argc, char *argv[])
{
    int i;
    // Uruchomienie watchdog-a, okres odświeżana 10 sek
    outp(WDOG_ENABLE,10);
    do {
        // Odświeżanie ---
        inp(WDOG_REFRESH);
        sleep(10);
    } while(1);
    inp(WDOG_DISABLE);
}

```

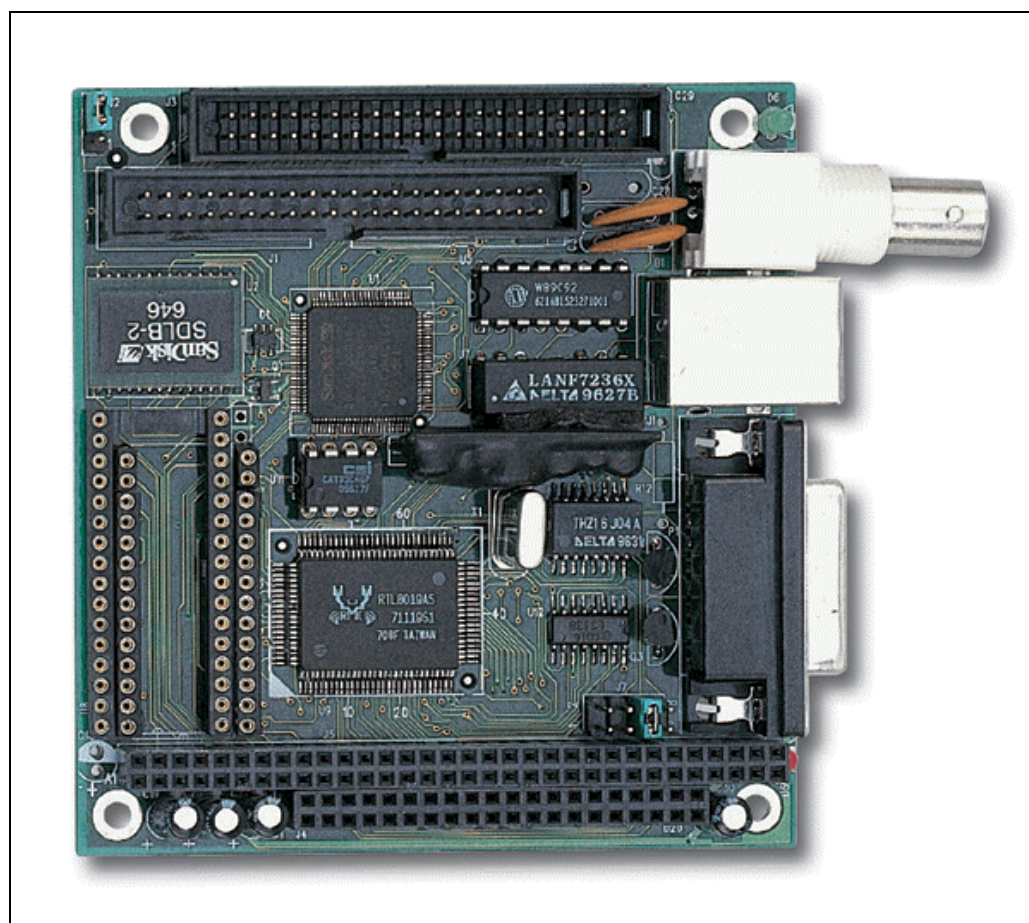
Program włączenie i odświeżania watchdoga

4.3 Moduł NC-812

Chipset	Realtek ALG 2032, szybkość 10MB, złącza BNC, UTP, AUI
Szyna systemowa	Złącze PC104
Gniazdo pamięci flash	32 bitowe gniazdo pamięci flash dla M-system Disk-On-Chip 2MB-144MB
Specyfikacja modułu NC-812	

Sterownik karty sieciowej jest zgodny z kartą NE2000. Obsługiwany jest przez sterownik Net.ether1000.

4.3.1.1 Net.ether1000 -p300 -l1 &



Wygląd modułu NC-812

5. Dyski półprzewodnikowe typu flash

Pamięci flash są rodzajem półprzewodnikowej pamięci nieulotnej. Pamięć flash jest odmianą pamięci EEPROM (Electrically Erasable Programmable Read-Only Memory). Kasowanie i zapis odbywa się przez podanie odpowiednich sygnałów elektrycznych. Nazwa "flash" pochodzi od możliwości bardzo szybkiego kasowania zapisanych komórek pamięci. Wszystkie pamięci flash zużywają się po pewnej liczbie cykli zapis / kasowanie.

Pamięci Flash produkuje się w dwóch formach: NOR i NAND. Nazwy odnoszą się do typu bramki logicznej używanej w każdej komórce pamięci.

Pamięć NOR

Jako pierwszą opracowano pamięć NOR w firmie Intel w roku 1988. Pamięci NOR umożliwiają swobodne adresowanie całego zakresu danych. Liczba cykli zapis / kasowanie 10 tys. – 1000 tys. W oparciu o te pamięci zbudowane są karty pamięciowe Compact Flash i Smart Media.

Pamięć NAND

Pamięci NAND opracowano w firmie Toshiba w roku 1989. W stosunku do pamięci NOR ma krótszy czas kasowania i zapisu, większą gęstość, mniejszy współczynnik kosztu na 1 bit oraz dziesięciokrotnie większą trwałość. Liczba cykli zapis / kasowanie 100 tys – 1mln. Dostęp do pamięci nie jest swobodny ale sekwencyjny. Na bazie pamięci NAND zbudowane są karty pamięciowe MMC, Secure Digital oraz Memory Stick.

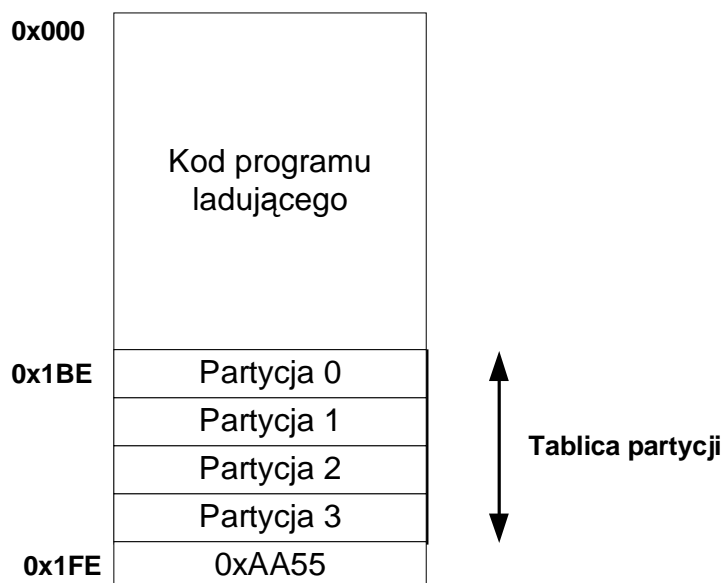
Dysk półprzewodnikowy NAND składa się z obszarów 512 bajtów (sektorów). Obszary te nie mogą być wprost zmieniane ale tylko w całości kasowane i od nowa zapisywane.

6. BIOS i ładowanie systemu operacyjnego

BIOS (ang. Basic Input Output System) zapewnia dostęp do podstawowych komponentów komputera.

Jego funkcje to:

1. Sprawdzenie integralności kodu programu BIOS
2. Ustalenie rozmiaru i sprawdzenie pamięci głównej
3. Znalezienie, zainicjalizowanie i skatalogowanie wszystkich magistrali
4. Dostarczenie interfejsu do konfiguracji systemu (BIOS Setup)
5. Zidentyfikowanie urządzeń zdolnych do wczytywania systemu
6. Uruchomienie programu ładującego system operacyjny (Użytkownik może określić z którego urządzenia (dysku) należy załadować system operacyjny gdy urządzeń jest więcej).
7. Załadowane z wybranego dysku pierwsze 512 bajtów do pamięci operacyjnej pod adres 0000:7C00.
8. Przekazanie sterowania do rozkazu znajdującego się pod adresem 0000:7C00.



Rysunek 1 Struktura sektora ładującego MBR

1	Flaga aktywności
2	Początek partycji
3	Typ partycji
4	Koniec partycji
5	Sektor początkowy partycji
6	Liczba sektorów partycji

Rysunek 2 Zawartość tablicy partycji

Jedna partycja oznaczona jest jako aktywna. Zawiera ona także program ładujący.

5 Instalacja systemu operacyjnego na dysku Flash IDE

Dysk Flash IDE posiada złącze 40 pin. Można go umieścić na płycie głównej zamiast taśmy do dysku IDE. Mimo że jest to dysk typu Flash jest on widziany przez BIOS i system operacyjny jako zwykły dysk IDE. Pojemność dysku – obecnie do ok. 4GB.



Instalacja systemu na dysku Flash IDE może się odbywać:

1. Bezpośrednio jak na dysku obrotowym
2. Poprzez instalację dysku Flash IDE jako drugiego dysku w systemie, skopiowanie systemu i niezbędnych plików a następnie usunięcie dysku pierwszego.

W przypadku 1 może się okazać że system się nie mieści na dysku gdyż standardowa instalacja obejmuje nie używane w systemie wbudowanym pliki jak np. pliki pomocy. Wtedy należy wybrać opcje 2.

Uwaga!

Dyski Flash posiadają ograniczoną liczbę zapisów (obecnie 1do10 mln). Stąd często aktualizowane pliki należy umieścić na RAM dysku.

Instalacja systemu QNX4 na dysku Flash IDE (początkowo dysk instalowany jako drugi a potem jako pierwszy)

1. Instalacja dysku w gnieździe IDE:

Należy włożyć dysk do gniazda IDE2 na płycie głównej i podłączyć linie zasilające a następnie Zrestartować komputer.

2. Sprawdzenie czy dysk jest akceptowany przez BIOS

Należy przejść do BIOS'u i sprawdzić czy dysk jest akceptowany przez BIOS. Powinien być widziany jako drugi dysk IDE.

3. Sprawdzenie czy dysk jest akceptowany przez proces drivera Frys.ide

Należy zrestartować komputer. Dysk Flash powinien być widziany jako /dev/hd1

3. Utworzenie partycji na flash dysku diskonchip

Partycję na nowym dysku tworzy się za pomocą standardowego programu fdisk. Można to także wykonać wsadowo:

```
# fdisk /dev/hd1 add -f 1 QNX ALL
# fdisk /dev/hd1 boot QNX
```

Rezultat można sprawdzić wykonując polecenie:

```
# fdisk /dev/hd1 show
```

_____OS_____	Start	End	_____Number_____	Size
Boot				
name type	Cylinder	Cylinder	Cylinders	Blocks
1. ----- (---)	-----	-----	-----	-----
2. ----- (---)	-----	-----	-----	-----
3. ----- (---)	-----	-----	-----	-----
4. QNX (77)	0	976	977 500192	244 MB *

4. Zamontowanie nowej partycji

Utworzoną partycję należy zamontować w systemie plików w katalogu /mnt/diskonchip używając polecenia mount:

```
# ls /dev
# mount -p /dev/hd1
```

Za pomocą polecenia # ls /dev można sprawdzić czy partycja zamontowała się właściwie. Gdy tak w katalogu dev powinien być widoczny plik specjalny /dev/hd1t77 reprezentujący nową partycję.

5. Zainicjowanie systemu plików na nowej partycji:

Na nowej partycji należy zainicjować system plików używając polecenia dinit jak niżej:

```
# dinit -h /dev/hd1t77
# dinit -hb /dev/hd1t77
```

6. Utworzenie katalogu /mnt/flash

Nowo utworzony na flash dysku system plików należy zamontować w jakimś miejscu bieżącego systemu plików. W tym celu utworzymy katalog /mnt/flash

```
# mkdir /mnt/ flash
```

7. Zamontowanie nowej partycji na flash dysku w systemie plików

```
# mount /dev/hd1t77 /mnt/ flash
```

Wynik operacji należy zweryfikować poleceniem:

```
# chkfsys -u /mnt/flash
```

Po wykonaniu powyższych operacji nowy flash dysk posiada zainicjowany system plików który widoczny jest w katalogu /mnt/ flash. Można tam kopiować pliki. Jednak nie wystartujemy systemu z tego dysku gdyż nie ma tam jeszcze systemu.

8 Kopiowanie obrazu systemu na dysk flash

Należy skopiować obraz systemu na dysk flash do katalogu / jak podano poniżej:

```
# cp .boot /mnt/flash/.boot
```

9. Kopiowanie niezbędnych plików na flash dysk

Należy skopiować potrzebne pliki na flash dysk używając przygotowanego wcześniej skryptu copy2flash. Następnie należy skopiować drivery urządzeń używane w wbudowanym systemie:

```
# cp /bin/Dev32 /mnt/flash/bin/Dev32
# cp /bin/Dev32.ansi /mnt/flash/bin/Dev32.ansi
# cp /bin/Dev32.par /mnt/flash/bin/Dev32.par
# cp /bin/Dev32.ptty /mnt/flash/bin/Dev32.ptty
# cp /bin/Dev32.ser /mnt/flash/bin/Dev32.ser

# cd /mnt/flash/bin
# ln -s Dev32 Dev
# ln -s Dev32.ansi Dev.ansi
```

```
# ln -s Dev.ansi    Dev.con
# ln -s Dev.ansi    Dev.ditto
# ln -s Dev32.par   Dev.par
# ln -s Dev32.ptty  Dev.ptty
# ln -s Dev32.ser   Dev.ser
```

9 Przygotowanie pliku inicjacji systemu sysinit dla systemu docelowego
W pliku sysinit umieścić należy polecenia konfiguracji systemu i startowania aplikacji. Przykładowy plik `sysinit.1` pokazano poniżej.

```
set -i
export TZ=utc00
export TERM=qnx
rtc -l hw
Dev &
Dev.ansi -Q -n 4 &
reopen //0/dev/con1
Dev.ser &
Dev.par &
Dev.ptty &
Pipe &
emu87 &
tinit -T /dev/con* /dev/ser1 -t /dev/con1 &
# Uruchomienie driverów sieci -----
Net &
Net.ether1000 -p 300 -l1 -v &
nameloc &
netmap -f
# Start sieci TCP/IP -----
/usr/ucb/Socklet flash1 &
/usr/ucb/ifconfig en1 flash1 up
/usr/ucb/ifconfig lo0 localhost up
/usr/ucb/inetd /etc/config/inetd.1
# Start aplikacji -----
mój_prog1 &
...
```

Zawartość pliku sysinit.1

W pliku sysinit umieszczono polecenie:

```
tinit -T /dev/con* /dev/ser1 -t /dev/con1 &
```

Polecenie to inicjuje konsolę, w tym ma porcie szeregowym /dev/ser1.

Jeżeli przyłączymy do tego portu terminal szeregowy przy pomocy kabla

typu null modem. Parametry programu terminalowego ustawiamy jak poniżej:

Baud	9600
Bits	8
Stopbits	1
Parity	brak
Handshaking	brak

10 Konfiguracja i uruchomienie sieci QNX Net

Wygodnie jest konfigurować nowy komputer przez sieć systemu QNX. W tym celu należy:

- Wgrać licencje do pliku /etc/licenses (za pomocą polecenia licenses)
- Utworzyć plik /etc/config/netmap zgodnie z regułami tworzenia tego pliku.

Jeżeli karta sieciowa nowego komputera ma numer 0005 8A00 02BA i będzie on węzłem 1 naszej sieci QNX Net to plik netmap może być jak poniżej.

1	1	00C0	DDF7	D782
2	1	0005	8A00	02BA

10. Konfiguracja i uruchomienie sieci TCP/IP

Sieć TCP/IP startowana jest w pliku sysinit. Nowy węzeł nosi nazwę flash1. Jego numer IP powinien być podany w pliku /etc/hosts. Plik ten powinien zawierać linię na przykład jak poniżej:

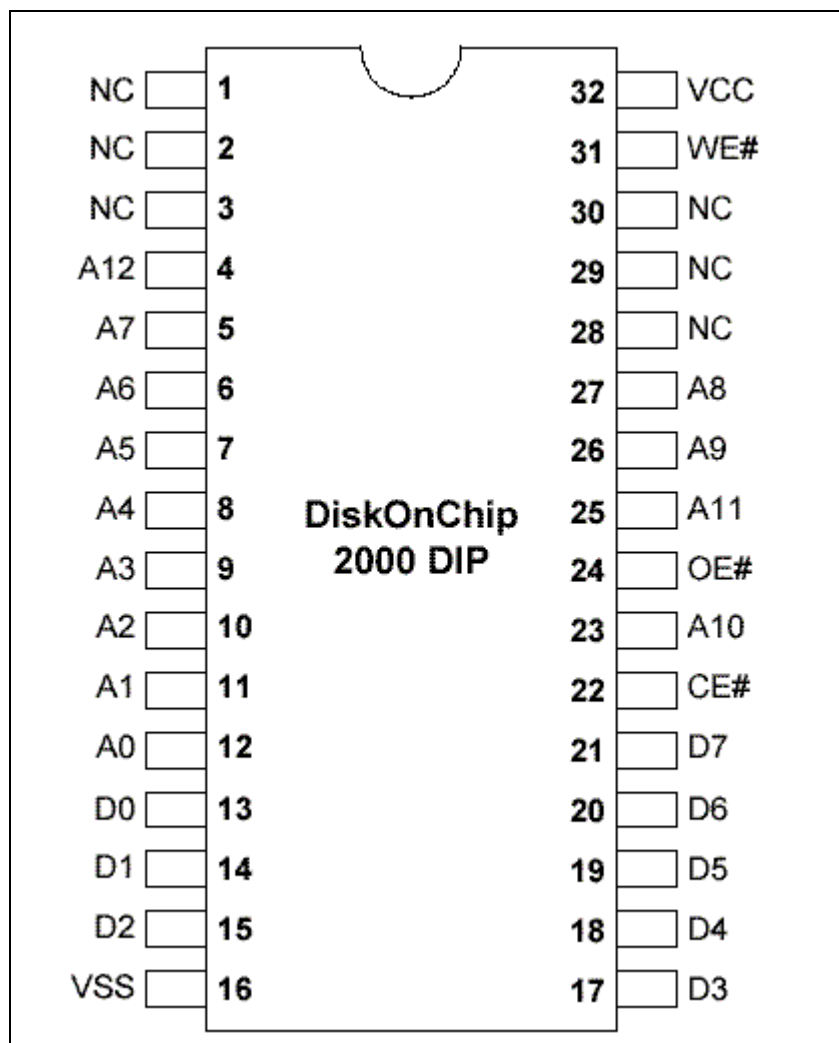
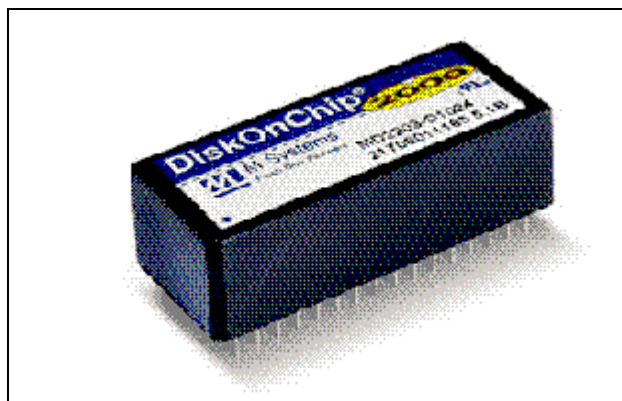
```
192.168.0.225    flash1
```

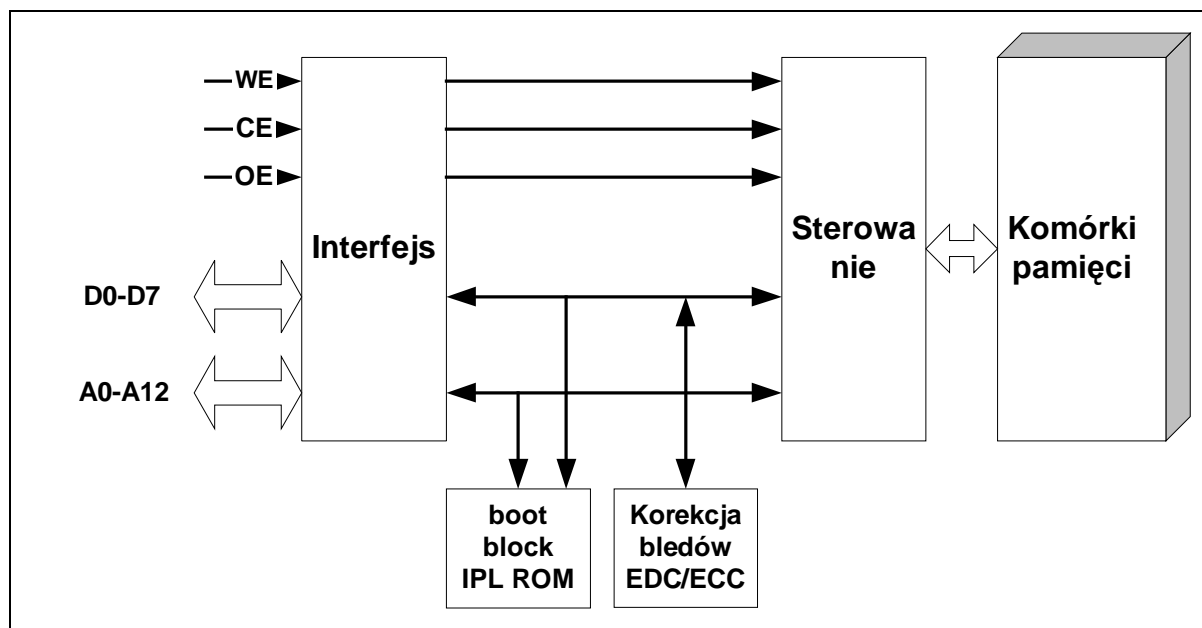
11. Uruchomienie programów aplikacyjnych

Programy aplikacyjne umieszczamy w odpowiednich katalogach i uruchamiamy podając ich nazwy w pliku sysinit.1.

6 Budowa i działanie pamięci DiskOnChip 2000

Pamięć DiskOnChip widziana jest jako 8 KB obszar w przestrzeni adresowej pamięci (*ang. Memory Mapped Device*)





Uproszczony schemat blokowy układu DiskOnChip2000

6.1.1.1.1

6.1.1.1.2 Interfejs modułu

Zapewnia połączenie pomiędzy standardowymi sygnałami magistrali a układami wewnętrznymi pamięci.

6.1.1.1.3 Boot Block

Obszar pamięci zawierający kod uruchamiany przez BIOS podczas procedury wyszukiwania urządzeń dodatkowych. Kod jest uruchamiany przez BIOS i ładuje do RAM SPL (*ang. Secondary Program Loader*).

6.1.1.1.4 Moduł korekcji błędów

Moduł obsługujący kod detekcyjny Reeda Salomona (*ang. Reed-Solomon Error Detection Code (EDC)*). Automatycznie dołączany blok 6 bajtów do każdej strony pamięci (512 bajtów). Korekcja błędów (*ang. ECC Error Correction Code*) zapewniana jest przez driver systemu plików FFS.

6.1.1.1.5 Moduł sterowania

Moduł sterowania zapewnia przekazywanie adresów i danych pomiędzy modułem interfejsowym a układami pamięci.

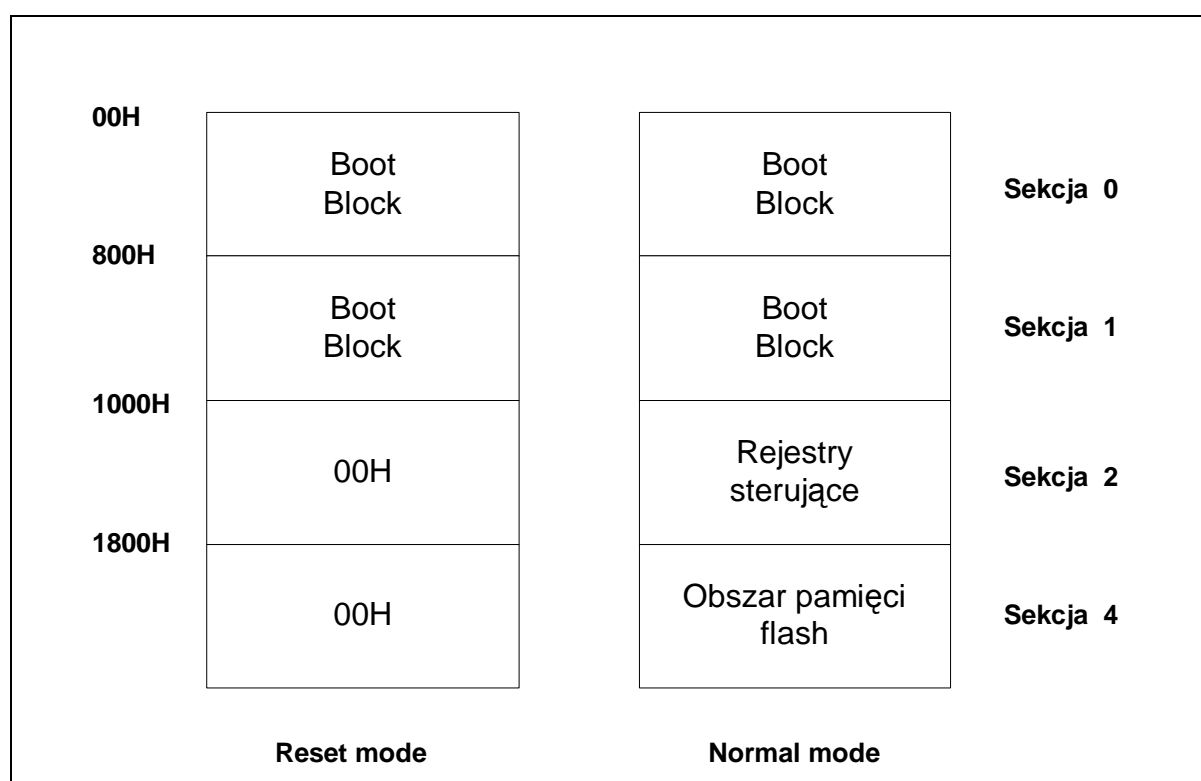
6.1 Tryby pracy

Tryby pracy

Tryb normalny (*ang. Normal Mode*) – układ reaguje na sygnały odczytu i zapisu

Tryb startowy (*ang. Reset Mode*) - układ reaguje tylko na sygnały odczytu.

Przełączenie trybu pracy następuje gdy nastąpi prawidłowa sekwencja zapisu do rejestrów sterowania.



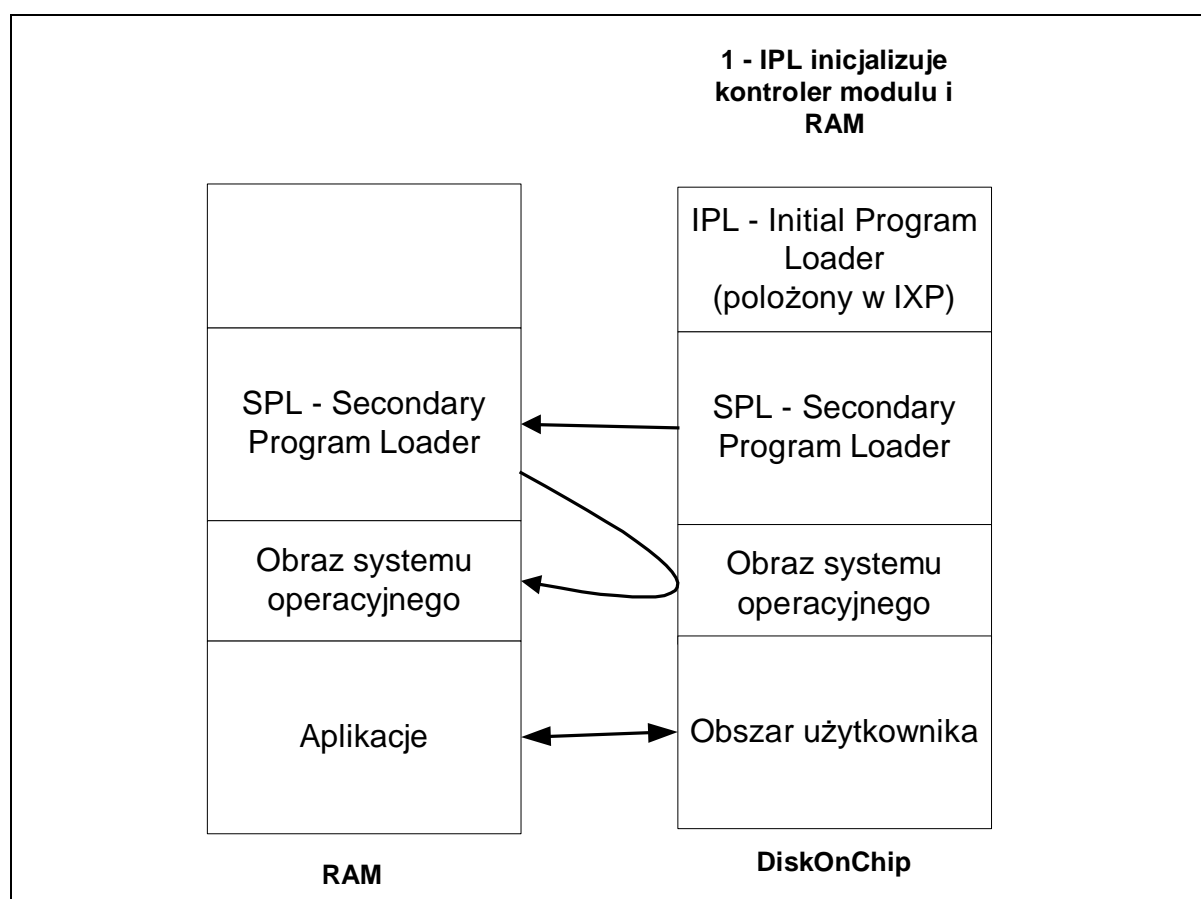
Tryby pracy systemu z DiskOnChip

6.2 Własności Ładowanie systemu z pamięci DiskOnCip

Przy starcie BIOS wykonuje procedurę POST (*ang. Power On Self-Test*). Przeszukiwany jest obszar pomiędzy adresem 0xC8000H a 0xEFFFFH w poszukiwaniu urządzeń rozszerzających. Gdy dysk jest zainstalowany znajdująca się sygnatura urządzenia i wykonywany kod zawarty w IPL (*ang. Initial Program Loader*). IPL ładuje do RAM drivera powodujące że pamięć flash widziana jest jako zwykły dysk i przyporządkowuje mu literę. Następnie sterowanie przekazywane jest do BIOS'a.

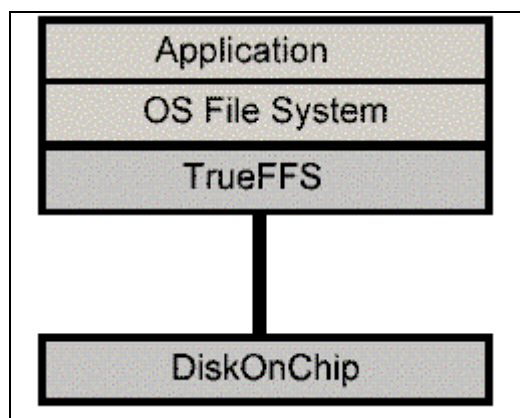
Gdy DiskOnChip jest jedynym dyskiem w systemie to będzie on oznaczony jako C i z niego będzie przeprowadzone ładowanie systemu.

Aby system znalazł się na dysku należy go tam załadować. W systemie MS-DOS wykonuje się to procedurą SYS C:



6.3 TrueFFS (True Flash File System)

Standard systemu plików dla pamięci flash wprowadzony przez firmę M-system w 1992 roku. Jest to warstwa oprogramowania pomiędzy systemem operacyjnym a pamięcią flash.



Zapewnia następujące funkcje:

1. Dostarcza driverów zapewniających że flash dysk widziany jako zwykłe urządzenie blokowe. Dostarczane jest API zapewniające odczyt i zapis sektorów jak w zwykłym dysku.
2. Zapewnia zarządzanie uszkodzonymi blokami. W pamięci może wystąpić pewna liczba uszkodzonych bloków które są wyłączane z użycia.
3. Zapewnia wykrywanie i korekcję błędów w oparciu o kody korekcyjne
4. Zwiększa trwałość urządzenia poprzez równomierne rozłożenie zapisów. Częste zapisy kierowane są do różnych bloków.
5. Zapewnia bezpieczeństwo danych w przypadku wyłączenia zasilania podczas zapisu.

TrueFSS dostarczany dla systemów:

Windows CE/NT/NT Embedded/XP, Linux (różne jądra), VxWorks, Nucleus, QNX, DOS, Symbian i inne.

7 Instalacja systemu QNX4 na dysku półprzewodnikowym Millennium Plus

6.1.1.1.6 Narzędzia dostarczane przez producenta flash dysku

Firma m-system (<http://www.m-sys.com>) dostarcza narzędzia potrzebne do utworzenia aplikacji wbudowanej działającej po kontrolą systemu QNX4. Są to następujące pliki i programy:

1	<code>readme.txt</code>	Plik zawierający instrukcję instalacji
2	<code>Fsys.diskonchip</code>	Driver flash dysku
3	<code>dformat</code>	Narzędzie do formatowania dysku
4	<code>copy2doc</code>	Skrypt kopiowania plików do systemu docelowego
5	<code>doc2000</code>	Makefile do sporządzania obrazu systemu QNX4.25 dla systemu docelowego
6	<code>include</code>	Katalog zawierający pliki nagłówkowe

6.1.1.1.7 Przygotowanie dysku w systemie QNX

1. Formatowanie dysku:

Formatowanie dysku wykonuje się przy pomocy narzędzia `dformat`.

```
# dformat
```

Jako opcję `-w:address` podać można adres początku okna dysku. Na przykład

```
# format -w:d8000
```

Komputer PM-1041 umożliwia następujące ustawienie (za pomocą zworek) adresu początkowego okna dysku.

1	C8000– C9FFF
2	D0000– D1FFF
3	D8000-D9FFF

2. Uruchomienie drivera dysku `diskonchip`

Aby uzyskać dostęp do flash dysku należy uruchomić jego driver.

Wykonuje się to za pomocą polecenia:

```
# Fsys.diskonchip &
```

W katalogu /dev powinien pojawić się plik specjalny /dev/tffs0 lub /dev/tffs0.0 reprezentujący nowe urządzenie. Można to sprawdzić poleceniem:

```
# ls /dev/tffs*
```

Opcje drivera można rozpoznać wykonując standardowe polecenie use:

```
# use Fsys.diskonchip
```

3. Utworzenie partycji na flash dysku diskonchip

Partycję na nowym dysku tworzy się za pomocą standardowego programu fdisk. Można to także wykonać wsadowo:

```
# fdisk /dev/tffs0 add -f 1 QNX ALL
```

```
# fdisk /dev/tffs0 boot QNX
```

Rezultat można sprawdzić wykonując polecenie:

```
# fdisk /dev/tffs0 show
```

4. Zamontowanie nowej partycji

Utworzoną partycję należy zamontować w systemie plików w katalogu /mnt/diskonchip używając polecenia mount:

```
# ls /dev/tffs*
```

```
# mount -p /dev/tffs0
```

Za pomocą polecenia # ls /dev/tffs* można sprawdzić czy partycja zamontowała się właściwie. Gdy tak w katalogu dev powinien być widoczny plik specjalny /dev/tffs0t77 reprezentujący nową partycję.

5. Zainicjowanie systemu plików na nowej partycji:

Na nowej partycji należy zainicjować system plików używając polecenia dinit jak niżej:

```
# dinit -h /dev/tffs0t77
```

```
# dinit -hb /dev/tffs0t77
```

6. Utworzenie katalogu /mnt/diskonchip

Nowo utworzony na flash dysku system plików należy zamontować w jakimś miejscu bieżącego systemu plików. W tym celu utworzymy katalog /mnt/diskonchip_

```
# mkdir /mnt/diskonchip
```

7. Zamontowanie nowej partycji na flash dysku w systemie plików

```
# mount /dev/tffs0t77 /mnt/diskonchip
```

Wynik operacji należy zweryfikować poleceniem:

```
# chkfsys -u /mnt/diskonchip
```

Po wykonaniu powyższych operacji nowy flash dysk posiada zainicjowany system plików który widoczny jest w katalogu /mnt/diskonchip. Można tam kopiować pliki. Jednak nie wystartujemy systemu z tego dysku gdyż nie ma tam jeszcze systemu.

6.1.1.1.7.1 Przygotowanie ładowalnej wersji systemu

1. Edycja pliku definicji systemu

Zmienić katalog bieżący na /boot. Do katalogu /boot/build należy skopiować plik Makefile o nazwie doc2000 definiujący obraz systemu docelowego. Jeżeli wystąpi taka potrzeba plik można poddać modyfikacji. W poniższym przykładzie wygenerowano system dla węzła 2 podając procesowi Proc32 opcję -l 2.

```
/boot/sys/boot
$ /boot/sys/boot -v

/boot/sys/Proc32
$ /boot/sys/Proc32 -l 2

/boot/sys/Slib32
$ /boot/sys/Slib32

/boot/sys/Slib16
$ /boot/sys/Slib16

/bin/Fsys
$ /bin/Fsys

/bin/Fsys.diskonchip
$ /bin/Fsys.diskonchip

/bin/mount
$ /bin/mount -p /dev/tffs0 /dev/tffs0t77 /
/bin/sinit
$ /bin/sinit TERM=qansi
```

Zawartość pliku doc2000

2. Przygotowanie obrazu systemu

Należy przygotować obraz systemu korzystając z systemowego narzędzia make:

```
# make b=doc2000
```

W katalogu /boot/image powinien powstać plik doc2000 i doc2000.map. Plik doc2000 jest obrazem systemu.

3. Kopiowanie obrazu systemu na dysk flash

Należy skopiować obraz systemu na dysk flash do katalogu / jak podano poniżej:

```
# cp /boot/images/doc2000 /mnt/diskonchip/.boot
```

4. Kopiowanie niezbędnych plików na flash dysk

Należy skopiować potrzebne pliki na flash dysk używając dostarczonego przez producenta dysku skryptu copy2doc.

```
# cd /usr/local/diskonchip
# chmod a+x copy2doc
# ./copy2doc
```

Następnie należy skopiować drivery urządzeń używane w wbudowanym systemie:

```
# cp /bin/Dev32 /mnt/diskonchip/bin/Dev32
# cp /bin/Dev32.ansi
/mnt/diskonchip/bin/Dev32.ansi
# cp /bin/Dev32.par
/mnt/diskonchip/bin/Dev32.par
# cp /bin/Dev32.ptty
/mnt/diskonchip/bin/Dev32.ptty
# cp /bin/Dev32.ser
/mnt/diskonchip/bin/Dev32.ser

# cd /mnt/diskonchip/bin
# ln -s Dev32 Dev
# ln -s Dev32.ansi Dev.ansi
# ln -s Dev.ansi Dev.con
# ln -s Dev.ansi Dev.ditto
# ln -s Dev32.par Dev.par
# ln -s Dev32.ptty Dev.ptty
# ln -s Dev32.ser Dev.ser
```

5. Przygotowanie pliku inicjacji systemu sysinit dla systemu docelowego

W pliku sysinit umieścić należy polecenia konfiguracji systemu i startowania aplikacji. Przykładowy plik `sysinit.2` pokazano poniżej.

```
set -i
export TZ=utc00
export TERM=qnx
rtc -l hw
Dev &
Dev.ansi -Q -n 4 &
```

```

reopen //0/dev/con1
Dev.ser &
Dev.par &
Dev.ptty &
Pipe &
emu87 &
tinit -T /dev/con* /dev/ser1 -t /dev/con1 &
# Uruchomienie driverów sieci -----
Net &
Net.ether1000 -p 300 -ll -v &
nameloc &
netmap -f
# Start sieci TCP/IP -----
/usr/ucb/Socklet pc104 &
/usr/ucb/ifconfig en1 pc104 up
/usr/ucb/ifconfig lo0 localhost up
/usr/ucb/inetd /etc/config/inetd.1
# Start aplikacji -----
mój_prog1 &
...

```

Zawartość pliku sysinit.2

W pliku sysinit umieszczono polecenie:

```
tinit -T /dev/con* /dev/ser1 -t /dev/con1 &
```

Polecenie to inicjuje konsolę, w tym ma port szeregowy /dev/ser1. Jeżeli przyłączymy do tego portu terminal szeregowy przy pomocy kabla typu null modem. Parametry programu terminalowego ustawiamy jak poniżej:

Baud	9600
Bits	8
Stopbits	1
Parity	brak
Handshaking	brak

6. Konfiguracja i uruchomienie sieci QNX Net

Wygodnie jest konfigurować nowy komputer przez sieć systemu QNX. W tym celu należy:

- Wgrać licencje do pliku /etc/licenses (za pomocą polecenia licenses)
- Utworzyć plik /etc/config/netmap zgodnie z regułami tworzenia tego pliku.

Jeżeli karta sieciowa nowego komputera ma numer 0005 8A00 02BA i będzie on węzłem 2 naszej sieci QNX Net to plik netmap może być jak poniżej.

1	1	00C0	DFE7	D782
2	1	0005	8A00	02BA

7. Konfiguracja i uruchomienie sieci TCP/IP

Sieć TCP/IP startowana jest w pliku sysinit. Nowy węzeł nosi nazwę pc104. Jego numer IP powinien być podany w pliku /etc/hosts. Plik ten powinien zawierać linię na przykład jak poniżej:

```
192.168.0.225    pc104
```

8. Uruchomienie programów aplikacyjnych

Programy aplikacyjne umieszczamy w odpowiednich katalogach i uruchamiamy podając ich nazwy w pliku sysinit.2.

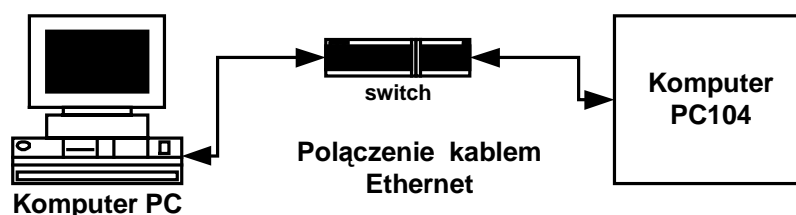
8 Testowanie połączenia z komputerem PC104

Połączenie z komputerem PC104 uzyskać można w następujący sposób:

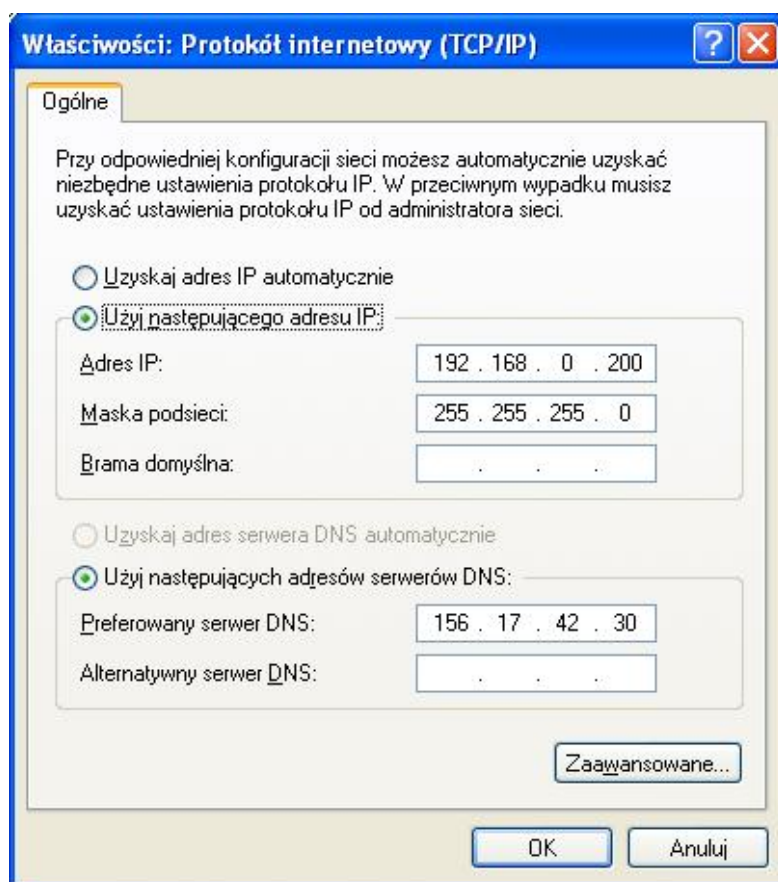
1. Wykorzystując połączenie sieciowe TCP/IP i program terminalowy telnet
2. Wykorzystując połączenie sieciowe QNX Net i drugi komputer pracujący pod kontrolą systemu QNX4
3. Wykorzystując port komunikacji szeregowej i program emulujący terminal (np. Hyperterminal z systemu Windows)

Testowanie połączenie sieciowego TCP/IP

Połączenie sieciowe testować można łącząc komputer PC104 do przełącznika sieciowego lub też używając sieciowego kabla skrzyżowanego.



W komputerze testującym należy ustawić adres IP i maskę sieci tak aby obydwa komputery znalazły się w jednej sieci. O ile używamy komputera pracującego pod kontrolą systemu Windows należy wybrać ikonę „Połączenie lokalne”, kliknąć prawym klawiszem myszki i ustawić właściwości połączenia TCP/IP jak poniżej.



Ekran 1 Ustawianie właściwości połączenie TCP/IP w komputerze testującym

Następnie należy otworzyć okno poleceń (Uruchom / cmd) i sprawdzić połączenie sieciowe używając polecenia:

```
C:> ping 192.168.0.225
```

Gdy połączenie jest sprawne należy uruchomić program telnet podając jako parametr adres IP komputera PC104:

```
C:> telnet 192.168.0.225
```

Po zgłoszeniu systemu należy się do niego zalogować



```

c:\ Telnet 192.168.0.225
Welcome to QNX 4.25
Copyright (c) QNX Software Systems Ltd. 1982,1998
login: root
Last login: Wed Jan 11 11:30:54 2006 on tty: not found
Wed Jan 11 11:37:35 2006
# ls /dev
con1          con6          null          ptyp3         tffs0t77
con2          con7          par1          ser1          ttyp0
con3          con8          ptyp0        ser2          ttyp1
con4          fd0           ptyp1        shmем        ttyp2
con5          kbd           ptyp2        tffs0         ttyp3
# _

```

Ekran 2 Zgłoszenie systemu QNX z komputera PC104

9 Literatura

- [1] PC104 Specification Version 2.5 November 2005, PC104 Embedded Consortium www.pc-104.org
- [2] Witryna firmy m-systems (<http://www.m-sys.com>)
- [3] Specyfikacja ISA - <http://www.techfest.com/hardware/bus/isa.htm>
- [4] Witryna firmy QUANTUM Korporacja Transferu Technologii <http://www.quantum.com.pl/>