

EM8247

Elfin Module Board

INSTRUKCJA OBSŁUGI
WERSJA 1.0

Spis treści

1	Wstęp	3
1.1	Moduł EM8247	3
1.2	Oprogramowanie	5
2	Sterowanie płytką	6
2.1	Interfejs WWW	6
2.2	Komunikacja z modułem za pomocą protokołu telnet	7
2.3	Komunikacja z modułem przez port szeregowy	7
2.4	System ratunkowy	7
3	Układ FPGA	8
3.1	Programowanie układu FPGA	8
3.2	Pseudopamięć	8
4	Płytką modułu	9
4.1	Wyprowadzone sygnały	9
4.2	Mapa pamięci FLASH	16
4.3	Pamięć EEPROM	17
5	Tworzenie własnego oprogramowania	18
5.1	Wymagania systemowe	18
5.2	Struktura katalogów	18
5.3	Instalacja kroskompilatora	19
5.4	Budowanie oprogramowania	19
5.5	Modyfikowanie systemu plików	20
5.6	Dodawanie własnych aplikacji	20
5.7	Wynikowy obraz systemu	20

Spis rysunków

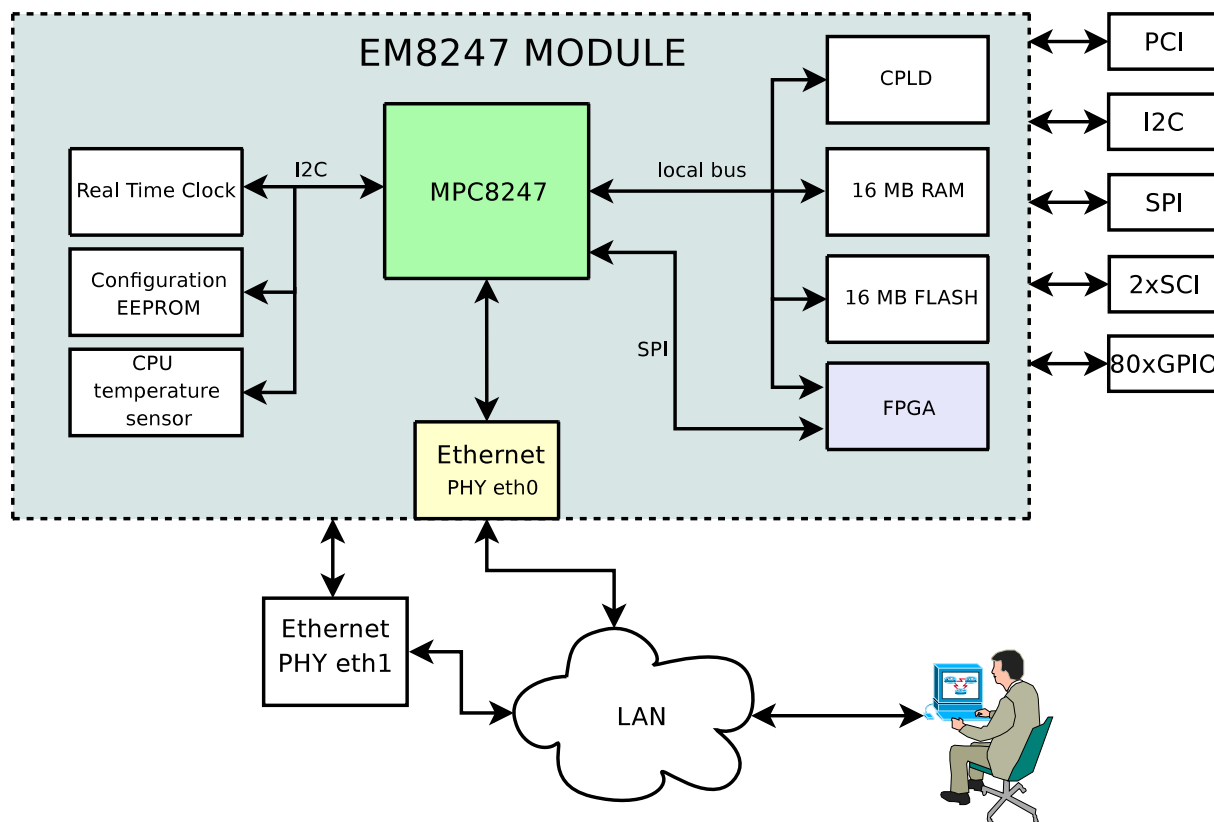
1.1	Struktura modułu EM8247	4
4.1	Mapa pamięci FLASH modułu EM8247	16

1.1 Moduł EM8247

Moduł *EM8247* („Elfin Module 8247”) jest mikrokomputerem ogólnego przeznaczenia opartym na procesorze MPC8247 (o architekturze PowerPC) firmy Freescale. Komunikacja z modułem odbywa się głównie poprzez sieć ethernetową, a konstrukcja pozwala na montowanie go w większych urządzeniach lub komputerach oraz dołączanie do niego zewnętrznych podzespołów. *EM8247* jest modułem niezależnym i posiada własny system operacyjny *Elfintosh* oparty na jądrze systemu *Linux*.

Strukturę modułu przedstawiono na rysunku 1.1. Jak wspomniano, „sercem” *EM8247* jest taktowany zegarem 266 MHz procesor MPC8247 (opcjonalnie dostępne są moduły z wyższymi częstotliwościami zegara). Jest to układ z rodziny PowerQUICC II, o rdzeniu PowerPC G2. Ponadto moduł zawiera następujące podzespoły:

- *16 MB pamięci Flash* zawierającej m.in. oprogramowanie urządzenia,
- *16 MB pamięci RAM* do ogólnego użytku,
- *układ CPLD* zawierający podstawową konfigurację procesora oraz udostępniający 16 pinów I/O,
- *układ FPGA Spartan-3* firmy Xilinx do dowolnego zastosowania,
- *zegar czasu rzeczywistego*,
- *pamięć EEPROM* do przechowywania dowolnych danych konfiguracyjnych (domyślnie zapisany w niej jest jedynie adres MAC modułu),
- *czujnik temperatury procesora*,
- *kontroler Ethernetu* do komunikacji z modułem (opcjonalnie dostępny jest także drugi kontroler).



Rysunek 1.1: Struktura modułu EM8247

Złącza wyprowadzone na zewnątrz modułu to:

- magistrala PCI,
- magistrala I²C,
- magistrala SPI,
- 2 porty szeregowo (SCI),
- 80 pinów I/O ogólnego przeznaczenia,
- złącze ethernetowe.

Dokładną organizację wyprowadzeń przedstawiono w rozdz. 4.1.

1.2 Oprogramowanie

Na całość zainstalowanego fabrycznie oprogramowania (wraz z systemem operacyjnym *Elfintosh*) składają się:

- bootloader *U-Boot*¹ w wersji 2009.11,
- zmodyfikowane jądro *Linuksa* w wersji 2.6.32,
- zbiór narzędzi *BusyBox*² w wersji 1.15.2,
- zestaw aplikacji firmy *Elfin*,
- serwer ze stroną WWW pozwalającą na sterowanie płytką,
- system ratunkowy (uruchamiany, jeżeli podstawowy system operacyjny zostanie uszkodzony przez użytkownika),
- opcjonalna biblioteka graficzna *QT Embedded*³ (przydatna przy podłączeniu zewnętrznego wyświetlacza graficznego LCD).

Wymienione powyżej oprogramowanie pozwala na:

- sterowanie podzespołami zawartymi w module *EM8247* lub dołączonymi do niego,
- programowanie zamieszczonego w module układu FPGA i obserwowanie działania załadowanego programu (patrz rozdz. 3),
- tworzenie własnych aplikacji lub wręcz modyfikowanie całego oprogramowania (dla zaawansowanych - patrz rozdz. 5).

¹<http://www.denx.de/wiki/UBoot>

²<http://www.busybox.net>

³<http://trolltech.com>

2.1 Interfejs WWW

Jednym ze składników oprogramowania modułu *EM8247* jest serwer ze stroną WWW, z którą można się połączyć podając do przeglądarki internetowej¹ adres **10.0.7.78**. Operacje udostępnione przez stronę uruchamia się klikając na umieszczone po lewej stronie przyciski, a są to:

Wstęp - wyświetlenie powitania,

Programowanie FPGA - załadowanie pliku z programem do układu FPGA (patrz rozdz. 3.1),

Pseudopamięć FPGA - wyświetlenie zawartości pseudopamięci FPGA lub jej modyfikacja (patrz rozdz. 3.2),

Konsola - uruchomienie powłoki systemowej („konsoli WWW”) w oknie przeglądarki²,

Uaktualnij firmware - załadowanie nowego firmware’u (obrazu całości oprogramowania) do pamięci Flash modułu,

Informacja o wersji - wyświetlenie wersji i daty utworzenia aktualnego firmware’u,

Restart - restart modułu.

¹Zalecane jest stosowanie przeglądarki Firefox lub Opera.

²Konsola WWW posiada ograniczoną funkcjonalność, pozwala jednak na wykonywanie podstawowych poleceń bez potrzeby logowania się do systemu.

2.2 Komunikacja z modułem za pomocą protokołu telnet

Głównym sposobem kontrolowania systemu *Elfintosh* jest zalogowanie się do niego (na adres **10.0.7.78**) poprzez protokół telnet. Aby tego dokonać w systemach uniksopodobnych (np. *Linux* lub *BSD*) wystarczy wpisać w linii poleceń komendę:

```
telnet 10.0.7.78
```

Przy zapytaniu o nazwę użytkownika należy podać `root`, system przyjmie użytkownika bez pytania o hasło.

System *Elfintosh* udostępnia konsolę `sh`, informacje na temat pracy w tej konsoli można znaleźć m.in. pod adresem <http://partmaps.org/era/unix/shell.html>.

2.3 Komunikacja z modułem przez port szeregowy

Moduł *EM8247* umożliwia łączenie się z konsolą systemu *Elfintosh* również za pomocą pierwszego portu szeregowego (w tym celu można wykorzystać choćby program `minicom` dla systemów uniksopodobnych). Ustawienia portu do korzystania z konsoli to:

- prędkość połączenia 115200 bps,
- 8 bitów danych,
- 1 bit stopu,
- brak kontroli parzystości.

Konsola na porcie szeregowym ma zastosowanie serwisowe, pozwala bowiem również kontrolować działanie bootloadera *U-Boot*.

2.4 System ratunkowy

Jeżeli użytkownik zmodyfikuje system *Elfintosh* (patrz rozdz. 5) i nowa wersja okaże się na tyle błędna, że uniemożliwi pracę oraz kolejną aktualizację oprogramowania, istnieje możliwość uruchomienia *systemu ratunkowego*, który jest domyślną, fabryczną wersją *Elfintosha*.

Aby uruchomić moduł z systemem ratunkowym, należy zrestartować go za pomocą umieszczonego na płycie modułu przycisku *RESET* jednocześnie trzymając znajdujący się obok przycisk *RESCUE*. Następnie po załadowaniu systemu można np. zaktualizować za pomocą strony WWW firmware do bezbłędnej wersji.

3.1 Programowanie układu FPGA

Moduł *EM8247* umożliwia proste programowanie układu FPGA za pomocą danych zapisanych w plikach typu `.bit` lub `.rbt`. Są to domyślne formaty plików programujących generowanych przez darmowe oprogramowanie *ISE WebPACK*¹ firmy Xilinx.

Wygenerowany plik programujący można załadować do FPGA za pomocą przycisku „Programowanie FPGA” na stronie WWW (patrz rozdz. 2.1) lub za pomocą polecenia konsoli:

```
programfpga nazwa_pliku
```

przy łączeniu się z modułem za pomocą protokołu telnet (patrz rozdz. 2.2). W drugim przypadku plik programujący musi znajdować się w systemie plików *Elfintosh*. Modyfikowanie systemu plików opisano w rozdziale 5.5.

3.2 Pseudopamięć

Układ FPGA „widziany” jest przez procesor jako obszar pamięci, której zawartość jest zmienna i zależy od programu załadowanego do układu. Ten obszar nazwano *pseudopamięcią* i można go obejrzeć za pomocą strony WWW lub wpisując w linii poleceń (po połączeniu protokołem telnet) komendę:

```
dumpfpgamem [adres]
```

Jeżeli jako argument powyższego polecenia poda się adres, to wyświetlona zostanie zawartość bajtu pod tym adresem (w przypadku braku argumentu polecenie wyświetli zawartość całej pseudopamięci).

Aby ustawić bajt pseudopamięci znajdujący się pod zadany adresem na zadaną wartość, należy skorzystać z interfejsu WWW lub użyć polecenia:

```
setfpgamem adres wartość
```

¹http://www.xilinx.com/ise/logic_design_prod/webpack.htm

4.1 Wyprowadzone sygnały

Do komunikacji z podłączonymi urządzeniami służą modułowi *EM8247* dwa złącza: IO oraz PCI. Funkcje pinów tych złącz opisane są poniżej. Oznaczenia typów pinów to:

O - pin wyjściowy,

I - pin wejściowy,

IO - pin wejściowo-wyjściowy,

P - pin zasilania,

OC - pin typu *open-collector*,

NC - pin nie używany.

Złącze IO (A)

Pin	Nazwa	Typ	Opis funkcji
A_1	GND	P	Masa zasilania
A_2	ETH0+O	O	Dodatnie wyjście ethernetu na transformator (zwarte z B_2)
A_3	ETH0-O	O	Ujemne wyjście ethernetu na transformator (zwarte z B_3)
A_4	ETH0_LEDYEL	O	Wyjście ethernetu sygnalizujące transmisję <i>ACK</i> (zwarte z B_4)
A_5	ETH0_LEDGRN	O	Wyjście ethernetu sygnalizujące połączenie <i>LINK</i> (zwarte z B_5)
A_6	ETH0+I	I	Dodatnie wejście ethernetu z transformatora (zwarte z B_6)
A_7	ETH0-I	I	Ujemne wejście ethernetu z transformatora (zwarte z B_7)
A_8	GND	P	Masa zasilania
A_9	RXD1	IO	Wejście SCI1 lub bit 4 portu <i>PC</i> procesora
A_10	TXD1	IO	Wyjście SCI1 lub bit 5 portu <i>PC</i> procesora

Pin	Nazwa	Typ	Opis funkcji
A_11	+3V3	P	Zasilanie 3,3V
A_12	TP2	NC	Pin testowy
A_13	FPGA_OUT	O	Wyjście programujące zewnętrzne układy FPGA
A_14	GND	P	Masa zasilania
A_15	IIC_SCL	O	Magistrala I ² C - linia zegara <i>SCL</i>
A_16	IIC_SDA	IO	Magistrala I ² C - linia danych <i>SDA</i>
A_17	+3V3	P	Zasilanie 3,3V
A_18	IO_INT#0	IO	Bit 14 portu <i>PC</i> procesora, przerwanie zewnętrzne INT#0 lub przerwanie PCI_INT
A_19	IO_INT#1	IO	Bit 15 portu <i>PC</i> procesora, przerwanie zewnętrzne INT#1 lub przerwanie PCI_INT
A_20	GND	P	Masa zasilania
A_21	CPLD_IO_0	IO	Uniwersalne wejście-wyjście układu CPLD
A_22	CPLD_IO_1	IO	Uniwersalne wejście-wyjście układu CPLD
A_23	CPLD_IO_2	IO	Uniwersalne wejście-wyjście układu CPLD
A_24	CPLD_IO_3	IO	Uniwersalne wejście-wyjście układu CPLD
A_25	CPLD_IO_4	IO	Uniwersalne wejście-wyjście układu CPLD
A_26	CPLD_IO_5	IO	Uniwersalne wejście-wyjście układu CPLD
A_27	CPLD_IO_6	IO	Uniwersalne wejście-wyjście układu CPLD
A_28	CPLD_IO_7	IO	Uniwersalne wejście-wyjście układu CPLD
A_29	+3V3	P	Zasilanie 3,3V
A_30	FPGA_IO_0	IO	Uniwersalne wejście-wyjście układu FPGA
A_31	FPGA_IO_1	IO	Uniwersalne wejście-wyjście układu FPGA
A_32	FPGA_IO_2	IO	Uniwersalne wejście-wyjście układu FPGA
A_33	FPGA_IO_3	IO	Uniwersalne wejście-wyjście układu FPGA
A_34	FPGA_IO_4	IO	Uniwersalne wejście-wyjście układu FPGA
A_35	FPGA_IO_5	IO	Uniwersalne wejście-wyjście układu FPGA
A_36	FPGA_IO_6	IO	Uniwersalne wejście-wyjście układu FPGA
A_37	FPGA_IO_7	IO	Uniwersalne wejście-wyjście układu FPGA
A_38	GND	P	Masa zasilania
A_39	FPGA_IO_16	IO	Uniwersalne wejście-wyjście układu FPGA
A_40	FPGA_IO_17	IO	Uniwersalne wejście-wyjście układu FPGA
A_41	FPGA_IO_18	IO	Uniwersalne wejście-wyjście układu FPGA
A_42	FPGA_IO_19	IO	Uniwersalne wejście-wyjście układu FPGA
A_43	FPGA_IO_20	IO	Uniwersalne wejście-wyjście układu FPGA
A_44	FPGA_IO_21	IO	Uniwersalne wejście-wyjście układu FPGA
A_45	FPGA_IO_22	IO	Uniwersalne wejście-wyjście układu FPGA
A_46	FPGA_IO_23	IO	Uniwersalne wejście-wyjście układu FPGA
A_47	+3V3	P	Zasilanie 3,3V
A_48	FPGA_IO_32	IO	Uniwersalne wejście-wyjście układu FPGA
A_49	FPGA_IO_33	IO	Uniwersalne wejście-wyjście układu FPGA
A_50	FPGA_IO_34	IO	Uniwersalne wejście-wyjście układu FPGA
A_51	FPGA_IO_35	IO	Uniwersalne wejście-wyjście układu FPGA
A_52	FPGA_IO_36	IO	Uniwersalne wejście-wyjście układu FPGA
A_53	FPGA_IO_37	IO	Uniwersalne wejście-wyjście układu FPGA
A_54	FPGA_IO_38	IO	Uniwersalne wejście-wyjście układu FPGA
A_55	FPGA_IO_39	IO	Uniwersalne wejście-wyjście układu FPGA

Pin	Nazwa	Typ	Opis funkcji
A_56	GND	P	Masa zasilania
A_57	FPGA_IO_48	IO	Uniwersalne wejście-wyjście układu FPGA
A_58	FPGA_IO_49	IO	Uniwersalne wejście-wyjście układu FPGA
A_59	FPGA_IO_50	IO	Uniwersalne wejście-wyjście układu FPGA
A_60	FPGA_IO_51	IO	Uniwersalne wejście-wyjście układu FPGA
A_61	FPGA_IO_52	IO	Uniwersalne wejście-wyjście układu FPGA
A_62	FPGA_IO_53	IO	Uniwersalne wejście-wyjście układu FPGA
A_63	FPGA_IO_54	IO	Uniwersalne wejście-wyjście układu FPGA
A_64	FPGA_IO_55	IO	Uniwersalne wejście-wyjście układu FPGA
A_65	+3V3	P	Zasilanie 3,3V
A_66	+3V3	P	Zasilanie 3,3V
A_67	+3V3	P	Zasilanie 3,3V
A_68	GND	P	Masa zasilania
A_69	GND	P	Masa zasilania
A_70	GND	P	Masa zasilania

Złącze IO (B)

Pin	Nazwa	Typ	Opis funkcji
B_1	GND	P	Masa zasilania
B_2	ETH0+0	O	Dodatknie wyjście ethernetu na transformator (zwarne z A_2)
B_3	ETH0-0	O	Ujemne wyjście ethernetu na transformator (zwarne z A_3)
B_4	ETH0_LEDYEL	O	Wyjście ethernetu sygnalizujące transmisję <i>ACK</i> (zwarne z A_4)
B_5	ETH0_LEDGRN	O	Wyjście ethernetu sygnalizujące połączenie <i>LINK</i> (zwarne z A_5)
B_6	ETH0+I	I	Dodatknie wejście ethernetu z transformatora (zwarne z A_6)
B_7	ETH0-I	I	Ujemne wejście ethernetu z transformatora (zwarne z A_7)
B_8	GND	P	Masa zasilania
B_9	TXD2	IO	Wejście SCI2 lub bit 9 portu <i>PA</i> procesora
B_10	RXD2	IO	Wyjście SCI2 lub bit 8 portu <i>PA</i> procesora
B_11	+3V3	P	Zasilanie 3,3V
B_12	PPC_PD30	IO	Bit 30 portu <i>PD</i> procesora
B_13	PPC_PD31	IO	Bit 31 portu <i>PD</i> procesora
B_14	GND	P	Masa zasilania
B_15	SPI_MISO	IO	Magistrala SPI - linia <i>MISO</i> lub bit 16 <i>PD</i> procesora
B_16	SPI_MOSI	IO	Magistrala SPI - linia <i>MOSI</i> lub bit 17 <i>PD</i> procesora
B_17	+3V3	P	Zasilanie 3,3V
B_18	SPI_CLK	IO	Magistrala SPI - linia <i>CLK</i> lub bit 18 <i>PD</i> procesora
B_19	SPI_SEL	IO	Magistrala SPI - linia <i>SEL</i> lub bit 19 <i>PD</i> procesora
B_20	GND	P	Masa zasilania
B_21	CPLD_IO_8	IO	Uniwersalne wejście-wyjście układu CPLD
B_22	CPLD_IO_9	IO	Uniwersalne wejście-wyjście układu CPLD
B_23	CPLD_IO_10	IO	Uniwersalne wejście-wyjście układu CPLD
B_24	CPLD_IO_11	IO	Uniwersalne wejście-wyjście układu CPLD
B_25	CPLD_IO_12	IO	Uniwersalne wejście-wyjście układu CPLD

Pin	Nazwa	Typ	Opis funkcji
B_26	CPLD_IO_13	IO	Uniwersalne wejście-wyjście układu CPLD
B_27	CPLD_IO_14	IO	Uniwersalne wejście-wyjście układu CPLD
B_28	CPLD_IO_15	IO	Uniwersalne wejście-wyjście układu CPLD
B_29	+3V3	P	Zasilanie 3,3V
B_30	FPGA_IO_8	IO	Uniwersalne wejście-wyjście układu FPGA
B_31	FPGA_IO_9	IO	Uniwersalne wejście-wyjście układu FPGA
B_32	FPGA_IO_10	IO	Uniwersalne wejście-wyjście układu FPGA
B_33	FPGA_IO_11	IO	Uniwersalne wejście-wyjście układu FPGA
B_34	FPGA_IO_12	IO	Uniwersalne wejście-wyjście układu FPGA
B_35	FPGA_IO_13	IO	Uniwersalne wejście-wyjście układu FPGA
B_36	FPGA_IO_14	IO	Uniwersalne wejście-wyjście układu FPGA
B_37	FPGA_IO_15	IO	Uniwersalne wejście-wyjście układu FPGA
B_38	GND	P	Masa zasilania
B_39	FPGA_IO_24	IO	Uniwersalne wejście-wyjście układu FPGA
B_40	FPGA_IO_25	IO	Uniwersalne wejście-wyjście układu FPGA
B_41	FPGA_IO_26	IO	Uniwersalne wejście-wyjście układu FPGA
B_42	FPGA_IO_27	IO	Uniwersalne wejście-wyjście układu FPGA
B_43	FPGA_IO_28	IO	Uniwersalne wejście-wyjście układu FPGA
B_44	FPGA_IO_29	IO	Uniwersalne wejście-wyjście układu FPGA
B_45	FPGA_IO_30	IO	Uniwersalne wejście-wyjście układu FPGA
B_46	FPGA_IO_31	IO	Uniwersalne wejście-wyjście układu FPGA
B_47	+3V3	P	Zasilanie 3,3V
B_48	FPGA_IO_40	IO	Uniwersalne wejście-wyjście układu FPGA
B_49	FPGA_IO_41	IO	Uniwersalne wejście-wyjście układu FPGA
B_50	FPGA_IO_42	IO	Uniwersalne wejście-wyjście układu FPGA
B_51	FPGA_IO_43	IO	Uniwersalne wejście-wyjście układu FPGA
B_52	FPGA_IO_44	IO	Uniwersalne wejście-wyjście układu FPGA
B_53	FPGA_IO_45	IO	Uniwersalne wejście-wyjście układu FPGA
B_54	FPGA_IO_46	IO	Uniwersalne wejście-wyjście układu FPGA
B_55	FPGA_IO_47	IO	Uniwersalne wejście-wyjście układu FPGA
B_56	GND	P	Masa zasilania
B_57	FPGA_IO_56	IO	Uniwersalne wejście-wyjście układu FPGA
B_58	FPGA_IO_57	IO	Uniwersalne wejście-wyjście układu FPGA
B_59	FPGA_IO_58	IO	Uniwersalne wejście-wyjście układu FPGA
B_60	FPGA_IO_59	IO	Uniwersalne wejście-wyjście układu FPGA
B_61	FPGA_IO_60	IO	Uniwersalne wejście-wyjście układu FPGA
B_62	FPGA_IO_61	IO	Uniwersalne wejście-wyjście układu FPGA
B_63	FPGA_IO_62	IO	Uniwersalne wejście-wyjście układu FPGA
B_64	FPGA_IO_63	IO	Uniwersalne wejście-wyjście układu FPGA
B_65	+3V3	P	Zasilanie 3,3V
B_66	+3V3	P	Zasilanie 3,3V
B_67	+3V3	P	Zasilanie 3,3V
B_68	GND	P	Masa zasilania
B_69	GND	P	Masa zasilania
B_70	GND	P	Masa zasilania

Złącze PCI (A)

Pin	Nazwa	Typ	Opis funkcji
A_1	GND	P	Masa zasilania
A_2	GND	P	Masa zasilania
A_3	GND	P	Masa zasilania
A_4	+3V3	P	Zasilanie 3,3V
A_5	+3V3	P	Zasilanie 3,3V
A_6	+3V3	P	Zasilanie 3,3V
A_7	ETH1_TX_CLK	IO	Ethernet <i>PPC_MAC1</i> lub bit 19 <i>PC</i> procesora
A_8	ETH1_CRS	IO	Ethernet <i>PPC_MAC1</i> lub bit 26 <i>PB</i> procesora
A_9	ETH1_TX_EN	IO	Ethernet <i>PPC_MAC1</i> lub bit 29 <i>PB</i> procesora
A_10	ETH1_TX_ER	IO	Ethernet <i>PPC_MAC1</i> lub bit 31 <i>PB</i> procesora
A_11	ETH1_TXD0	IO	Ethernet <i>PPC_MAC1</i> lub bit 22 <i>PB</i> procesora
A_12	ETH1_TXD1	IO	Ethernet <i>PPC_MAC1</i> lub bit 23 <i>PB</i> procesora
A_13	ETH1_TXD2	IO	Ethernet <i>PPC_MAC1</i> lub bit 24 <i>PB</i> procesora
A_14	ETH1_TXD3	IO	Ethernet <i>PPC_MAC1</i> lub bit 25 <i>PB</i> procesora
A_15	MII_MDIO	IO	Magistrala MII (komunikacja MAC z PHY)
A_16	MII_MDC	O	Magistrala MII (komunikacja MAC z PHY)
A_17	GND	P	Masa zasilania
A_18	PCI_INTA#	I	Magistrala PCI
A_19	+3V3	P	Zasilanie 3,3V
A_20	PCI_RST#	O	Magistrala PCI
A_21	+3V3	P	Zasilanie 3,3V
A_22	PCI_GNT#0	O	Magistrala PCI
A_23	GND	P	Masa zasilania
A_24	PCI_GNT#1	O	Magistrala PCI
A_25	+3V3	P	Zasilanie 3,3V
A_26	PCI_GNT#2	O	Magistrala PCI
A_27	GND	P	Masa zasilania
A_28	PCI_AD30	IO	Magistrala PCI
A_29	+3V3	P	Zasilanie 3,3V
A_30	PCI_AD28	IO	Magistrala PCI
A_31	PCI_AD26	IO	Magistrala PCI
A_32	GND	P	Masa zasilania
A_33	PCI_AD24	IO	Magistrala PCI
A_34	PCI_IDSEL	O	Magistrala PCI
A_35	+3V3	P	Zasilanie 3,3V
A_36	PCI_AD22	IO	Magistrala PCI
A_37	PCI_AD20	IO	Magistrala PCI
A_38	GND	P	Masa zasilania
A_39	PCI_AD18	IO	Magistrala PCI
A_40	PCI_AD16	IO	Magistrala PCI
A_41	+3V3	P	Zasilanie 3,3V
A_42	PCI_FRAME#	IO	Magistrala PCI

Pin	Nazwa	Typ	Opis funkcji
A_43	GND	P	Masa zasilania
A_44	GND	P	Masa zasilania
A_45	PCI_TRDY#	IO	Magistrala PCI
A_46	+3V3	P	Zasilanie 3,3V
A_47	PCI_STOP#	IO	Magistrala PCI
A_48	GND	P	Masa zasilania
A_49	PCI_PAR	IO	Magistrala PCI
A_50	PCI_AD15	IO	Magistrala PCI
A_51	+3V3	P	Zasilanie 3,3V
A_52	PCI_AD13	IO	Magistrala PCI
A_53	PCI_AD11	IO	Magistrala PCI
A_54	GND	P	Masa zasilania
A_55	PCI_AD9	IO	Magistrala PCI
A_56	GND	P	Masa zasilania
A_57	PCI_BE#0	IO	Magistrala PCI
A_58	+3V3	P	Zasilanie 3,3V
A_59	PCI_AD6	IO	Magistrala PCI
A_60	PCI_AD4	IO	Magistrala PCI
A_61	GND	P	Masa zasilania
A_62	PCI_AD2	IO	Magistrala PCI
A_63	PCI_AD0	IO	Magistrala PCI
A_64	GND	P	Masa zasilania
A_65	+3V3	P	Zasilanie 3,3V
A_66	+3V3	P	Zasilanie 3,3V
A_67	+3V3	P	Zasilanie 3,3V
A_68	GND	P	Masa zasilania
A_69	GND	P	Masa zasilania
A_70	GND	P	Masa zasilania

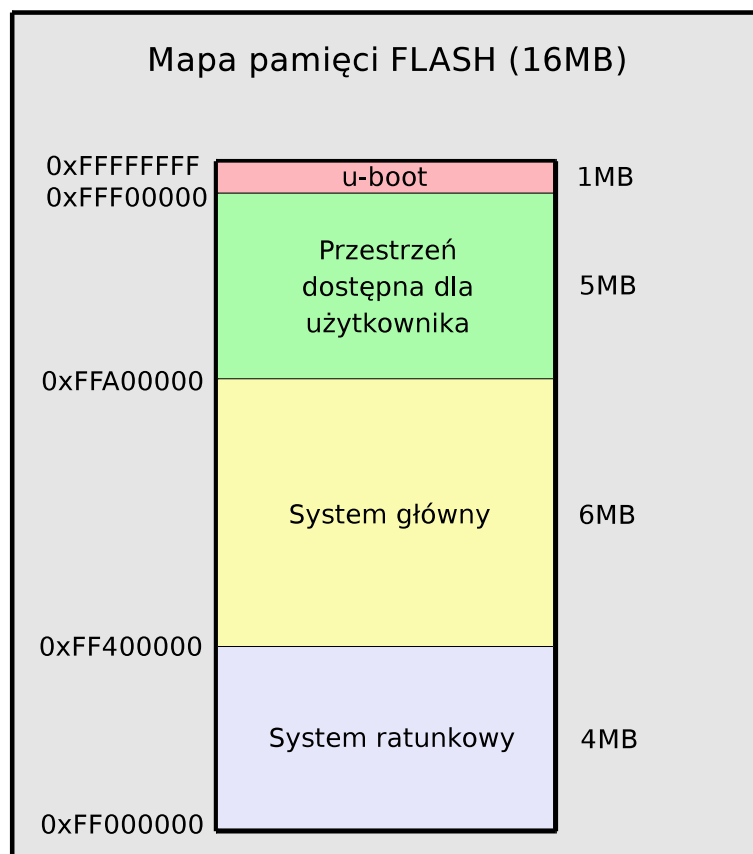
Złącze PCI (B)

Pin	Nazwa	Typ	Opis funkcji
B_1	GND	P	Masa zasilania
B_2	GND	P	Masa zasilania
B_3	GND	P	Masa zasilania
B_4	+3V3	P	Zasilanie 3,3V
B_5	+3V3	P	Zasilanie 3,3V
B_6	+3V3	P	Zasilanie 3,3V
B_7	ETH1_RX_CLK	IO	Ethernet <i>PPC_MAC1</i> lub bit 20 <i>PC</i> procesora
B_8	ETH1_COL	IO	Ethernet <i>PPC_MAC1</i> lub bit 27 <i>PB</i> procesora
B_9	ETH1_RX_DV	IO	Ethernet <i>PPC_MAC1</i> lub bit 30 <i>PB</i> procesora
B_10	ETH1_RX_ER	IO	Ethernet <i>PPC_MAC1</i> lub bit 28 <i>PB</i> procesora
B_11	ETH1_RXD0	IO	Ethernet <i>PPC_MAC1</i> lub bit 21 <i>PB</i> procesora
B_12	ETH1_RXD1	IO	Ethernet <i>PPC_MAC1</i> lub bit 20 <i>PB</i> procesora
B_13	ETH1_RXD2	IO	Ethernet <i>PPC_MAC1</i> lub bit 19 <i>PB</i> procesora

Pin	Nazwa	Typ	Opis funkcji
B_14	ETH1_RXD3	IO	Ethernet <i>PPC_MAC1</i> lub bit 18 <i>PB</i> procesora
B_15	MDINT#_RTH1	I	Wejście przerwania od PHY
B_16	LM75OUT	OC	Sygnalizacja stanu alarmowego temperatury procesora
B_17	GND	P	Masa zasilania
B_18	PCI_CLK	O	Magistrala PCI
B_19	GND	P	Masa zasilania
B_20	PCI_INTB#	I	Magistrala PCI
B_21	GND	P	Masa zasilania
B_22	PCI_REQ#0	I	Magistrala PCI
B_23	+3V3	P	Zasilanie 3,3V
B_24	PCI_REQ#1	I	Magistrala PCI
B_25	GND	P	Masa zasilania
B_26	PCI_REQ#2	I	Magistrala PCI
B_27	+3V3	P	Zasilanie 3,3V
B_28	PCI_AD31	IO	Magistrala PCI
B_29	PCI_AD29	IO	Magistrala PCI
B_30	GND	P	Masa zasilania
B_31	PCI_AD27	IO	Magistrala PCI
B_32	PCI_AD25	IO	Magistrala PCI
B_33	+3V3	P	Zasilanie 3,3V
B_34	PCI_BE#3	IO	Magistrala PCI
B_35	PCI_AD23	IO	Magistrala PCI
B_36	GND	P	Masa zasilania
B_37	PCI_AD21	IO	Magistrala PCI
B_38	PCI_AD19	IO	Magistrala PCI
B_39	+3V3	P	Zasilanie 3,3V
B_40	PCI_AD17	IO	Magistrala PCI
B_41	PCI_BE#2	IO	Magistrala PCI
B_42	GND	P	Masa zasilania
B_43	PCI_IRDY#	IO	Magistrala PCI
B_44	+3V3	P	Zasilanie 3,3V
B_45	PCI_DEVSEL#	IO	Magistrala PCI
B_46	GND	P	Masa zasilania
B_47	PCI_PERR#	IO	Magistrala PCI
B_48	+3V3	P	Zasilanie 3,3V
B_49	PCI_SERR#	IO	Magistrala PCI
B_50	+3V3	P	Zasilanie 3,3V
B_51	PCI_BE#1	IO	Magistrala PCI
B_52	PCI_AD14	IO	Magistrala PCI
B_53	GND	P	Masa zasilania
B_54	PCI_AD12	IO	Magistrala PCI
B_55	PCI_AD10	IO	Magistrala PCI
B_56	GND	P	Masa zasilania
B_57	PCI_AD8	IO	Magistrala PCI
B_58	PCI_AD7	IO	Magistrala PCI

Pin	Nazwa	Typ	Opis funkcji
B_59	+3V3	P	Zasilanie 3,3V
B_60	PCI_AD5	IO	Magistrala PCI
B_61	PCI_AD3	IO	Magistrala PCI
B_62	GND	P	Masa zasilania
B_63	PCI_AD1	IO	Magistrala PCI
B_64	GND	P	Masa zasilania
B_65	+3V3	P	Zasilanie 3,3V
B_66	+3V3	P	Zasilanie 3,3V
B_67	+3V3	P	Zasilanie 3,3V
B_68	GND	P	Masa zasilania
B_69	GND	P	Masa zasilania
B_70	GND	P	Masa zasilania

4.2 Mapa pamięci FLASH



Rysunek 4.1: Mapa pamięci FLASH modułu EM8247

4.3 Pamięć EEPROM

Na płytce *EM8247* znajduje się pamięć typu EEPROM. Przechowywany w niej jest adres MAC modułu, reszta pamięci pozostawiona jest do dyspozycji użytkownikowi. Zarządzanie pamięcią możliwe jest z poziomu powłoki systemu *Elfintosh* za pomocą polecenia

```
eeepromctl opcja [argument]
```

Możliwe wywołania tego polecenia, to:

```
eeepromctl -i nazwa_pliku - zapisanie zawartości pliku do pamięci,  
eeepromctl -o nazwa_pliku - zrzucenie zawartości pamięci do pliku,  
eeepromctl -s adres_mac - ustawienie nowego adresu MAC modułu,  
eeepromctl -g - wyświetlenie aktualnego adresu MAC.
```

Tworzenie własnego oprogramowania

5.1 Wymagania systemowe

Do modułu *EM8247* dołączone są kody źródłowe całego systemu *Elfintosh* oraz oprogramowanie pozwalające go modyfikować i tworzyć własne aplikacje. Wymienione oprogramowanie przeznaczone jest dla komputerów PC z zainstalowaną dystrybucją systemu operacyjnego *Linux*.

5.2 Struktura katalogów

Rozmieszczenie katalogów i plików systemu budowania oprogramowania dla modułu *EM8247* opisuje poniższa lista:

- CPLD* - pliki źródłowe programu układu CPLD,
- DOC* - dokumentacja modułu,
- DOC/src* - pliki źródłowe dokumentacji modułu,
- base_fs* - źródłowy system plików, tu skopiowane pliki będą dodane do systemu plików *Elfintosh*,
- bin* - wykonywalne pliki binarne, tu umieszczany jest kroskompilator po instalacji,
- busybox* - pliki źródłowe zestawu programów *BusyBox*,
- linux* - pliki źródłowe zmodyfikowanego jądra systemu *Linux*,
- root* - finalny system plików umieszczany w systemie *Elfintosh* (nie należy go modyfikować, jest on generowany na podstawie katalogu *base_fs* oraz dodanych aplikacji użytkownika),
- src* - pliki źródłowe kroskompilatora oraz podstawowego systemu plików (pliki te służą do generacji katalogów *base_fs* oraz *bin* podczas instalacji),
- tmp* - pliki tymczasowe,
- u-boot* - pliki źródłowe bootloadera *U-Boot*,
- qt* - pliki źródłowe biblioteki graficznej *QT Embedded*(<http://trolltech.com>),

<i>user</i>	- pliki dodawane i modyfikowane przez użytkownika,
<i>user/etc</i>	- katalog z plikiem <code>rc.user</code> zawierającym skrypt uruchamiany tuż po starcie systemu,
<i>user/scripts</i>	- dowolne skrypty powłoki (umieszczane docelowo w katalogu <code>/usr/bin</code> systemu plików <i>Elfintosh</i>),
<i>user/src</i>	- pliki źródłowe aplikacji użytkownika,
<i>user/www</i>	- pliki źródłowe strony www.

5.3 Instalacja kroskompilatora

Pierwszą czynnością jaką należy wykonać przed dalszymi operacjami jest jednorazowe zbudowanie i instalacja kroskompilatora. Aby rozpocząć ten proces, należy będąc w katalogu, w którym umieszczono pliki systemu budowania oprogramowania, wpisać polecenie:

```
make compiler
```

Następnie w celu wygenerowania źródłowego systemu plików i zbudowania pierwszego obrazu oprogramowania należy wpisać polecenie:

```
make
```

5.4 Budowanie oprogramowania

Wszystkich operacji budowania poszczególnych części oprogramowania dokonuje się za pomocą odpowiednich wywołań polecenia `make` w głównym katalogu systemu budowania. Dostępne wywołania, to:

- `make` - zbudowanie wszystkich części oprogramowania oprócz kroskompilatora,
- `make compiler` - zbudowanie i instalacja kompilatora,
- `make busybox` - zbudowanie zestawu aplikacji *BusyBox*,
- `make kernel` - zbudowanie jądra systemu *Linux*,
- `make uboot` - zbudowanie bootloadera,
- `make build_user` - zbudowanie aplikacji użytkownika,
- `make config_user` - skonfigurowanie zestawu aplikacji użytkownika,
- `make doc` - zbudowanie dokumentacji,
- `make qtembedded` - zbudowanie biblioteki QT Embedded.

Inne dostępne wywołania polecenia `make` to:

- `make menuconfig` - uruchomienie menu konfiguracyjnych dla jądra *Linuksa*, *BusyBoksa* oraz programów użytkownika,

- `make clean` - skasowanie wszystkich zbudowanych części oprócz kroskompilatora,
- `make clean_user` - skasowanie aplikacji użytkownika,
- `make clean_doc` - skasowanie plików wynikowych dokumentacji.

5.5 Modyfikowanie systemu plików

Źródłowy system plików znajduje się w katalogu *base_fs*. Umieszczone tam pliki po wywołaniu polecenia `make` znajdują się w analogicznych miejscach katalogu *root*, a po załadowaniu obrazu oprogramowania do modułu - w odpowiednich katalogach systemu plików *Elfintosh*.

W celu przywrócenia domyślnego źródłowego systemu plików należy skasować katalog *base_fs* i wywołać polecenie `make`, a wygenerowany zostanie nowy, domyślny *base_fs*.

5.6 Dodawanie własnych aplikacji

W katalogu *user/src* umieszczone są kody źródłowe dodatkowych aplikacji dla systemu *Elfintosh*. Aby dodać własną aplikację, należy utworzyć w tym miejscu nowy katalog i umieścić w nim kod źródłowy aplikacji oraz odpowiedni plik *Makefile*. Następnie będąc w głównym katalogu systemu budowania wywołać polecenie `make config_user` i - korzystając z menu - skonfigurować zestaw aplikacji użytkownika dodawanych do finalnego systemu operacyjnego. Przy następnym wywołaniu `make` lub `make build_user` podjęta zostanie próba kompilacji wybranych aplikacji oraz dodania ich do wynikowego pliku obrazu systemu (firmware).

5.7 Wynikowy obraz systemu

Po wywołaniu polecenia `make`, jeżeli wszystko się powiedzie, wygenerowany zostanie plik *ulmage* zawierający obraz właśnie zbudowanego systemu *Elfintosh* (tzw. firmware). Ten obraz może zostać załadowany do pamięci Flash za pomocą strony WWW (patrz rozdz. 2.1). Jeżeli zostanie to wykonane, to po restarcie modułu uruchomi się nowy, zbudowany przez użytkownika system operacyjny.