

# **Instrukcja do laboratorium Systemy wbudowane**

**ARM 9 TDMI**

(Wersja 2.4)

# 1. Regulamin pracy laboratorium

## a) Zaliczenie laboratorium

- Obecność na zajęciach jest obowiązkowa. Dopuszczalna jest jedna nieusprawiedliwiona nieobecność w ciągu semestru.
- Studenci wykonują zaplanowane ćwiczenia w grupach jednoosobowych.
- Każde ćwiczenie, za wyjątkiem ćwiczenia pierwszego, jest indywidualnie oceniane, z uwzględnieniem jakości i terminowości jego wykonania. Ocena i ew. uwagi są zapisywane w dzienniku.
- Każde pojedyncze ćwiczenie jest przewidziane do realizacji na jednych zajęciach (2 godziny), jeśli ćwiczenie jest przewidziane na dwa zajęcia, będzie ono oceniane na drugich zajęciach.
- Na ocenę zadania składa się: ocena programu (algorytm, sposób formalizowania kodu) oraz znajomość teoretyczna zagadnień wykorzystanych w programie.
- W przypadku stwierdzenia nieprzygotowania do zajęć, student jest z nich usuwany. Jest to równoznaczne z nieobecnością nieusprawiedliwioną.
- Ocena końcowa z laboratorium jest średnią ocen otrzymanych za poszczególne ćwiczenia.
- Rażące spóźnienie na zajęcia powoduje niedopuszczenie do ich odbycia.

## b) Kategorycznie zabrania się:

- Zdejmowania obudów komputerów.
- Podłączania i odłączania jakichkolwiek przewodów bez zezwolenia osoby prowadzącej zajęcia.
- Wynoszenia jakichkolwiek przedmiotów (w tym dokumentacji technicznej) znajdujących się w laboratorium.
- Przepisywania lub kopiowania programów, procedur lub funkcji od innych grup lub osób.
- Przebywania w laboratorium osobom nieodbywającym zajęć i nie związanych bezpośrednio z obsługą laboratorium.
- Spożywania posiłków i picia napojów w sąsiedztwie sprzętu komputerowego.

## c) W przypadku naruszenia zakazów wymienionych w pkt. b, na studentów mogą być nałożone sankcje, m. in.:

- Usunięcie z zajęć i wstawienie oceny niedostatecznej.
- Poniesienie kosztów ew. naprawy, jeżeli uszkodzenie powstało wskutek umyślnego działania studenta.

## 2. Opis sprzętu znajdującego się w laboratorium

W laboratorium znajduje się następujący sprzęt:

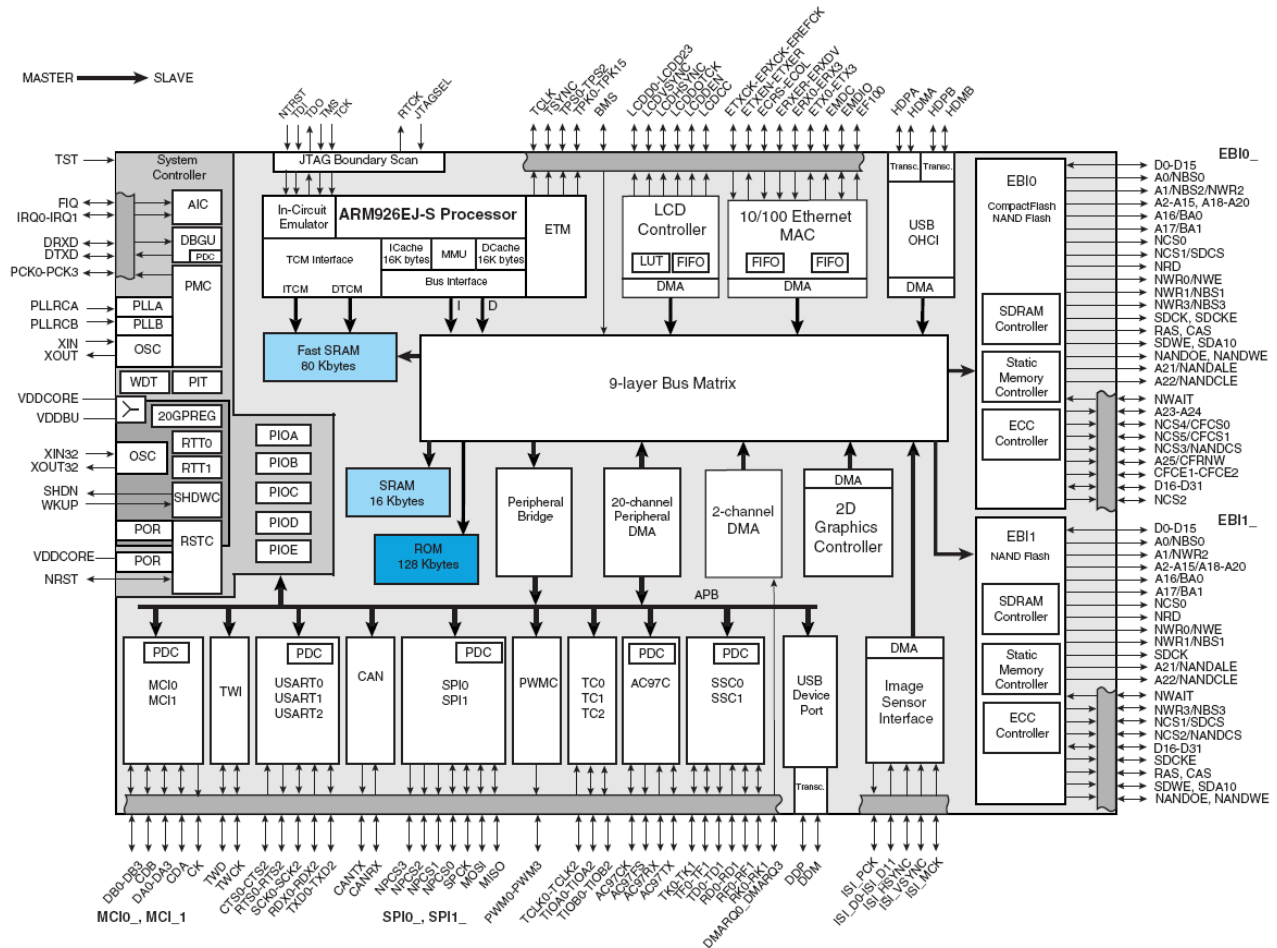
1. Komputery klasy PC.
2. Zestawy uruchomieniowe AT91SAM9263-EK Evaluation Board z 32-bitowymi procesorami AT91SAM9263 z rdzeniem ARM9TDMI.
3. Płytki rozszerzające z enkoderem, termometrem oraz wyświetlaczem LED.
4. Przenośne stabilizowane zasilacze 12 V/1 A.
5. Adaptery JTAG łączące kanały diagnostyczne procesorów ARM z komputerem PC.
6. Komplet przewodów połączeniowych (EIA232, USB) .

Zestawy uruchomieniowe składają się z dwóch modułów:

1. Płyty bazowej z procesorem ARM,
2. Modułu z wyświetlaczem, enkoderem i termometrem

Na rysunku 1 zamieszczono schemat blokowy procesora AT91SAM9263. Procesor wyposażony jest w następujące pamięci wewnętrzne:

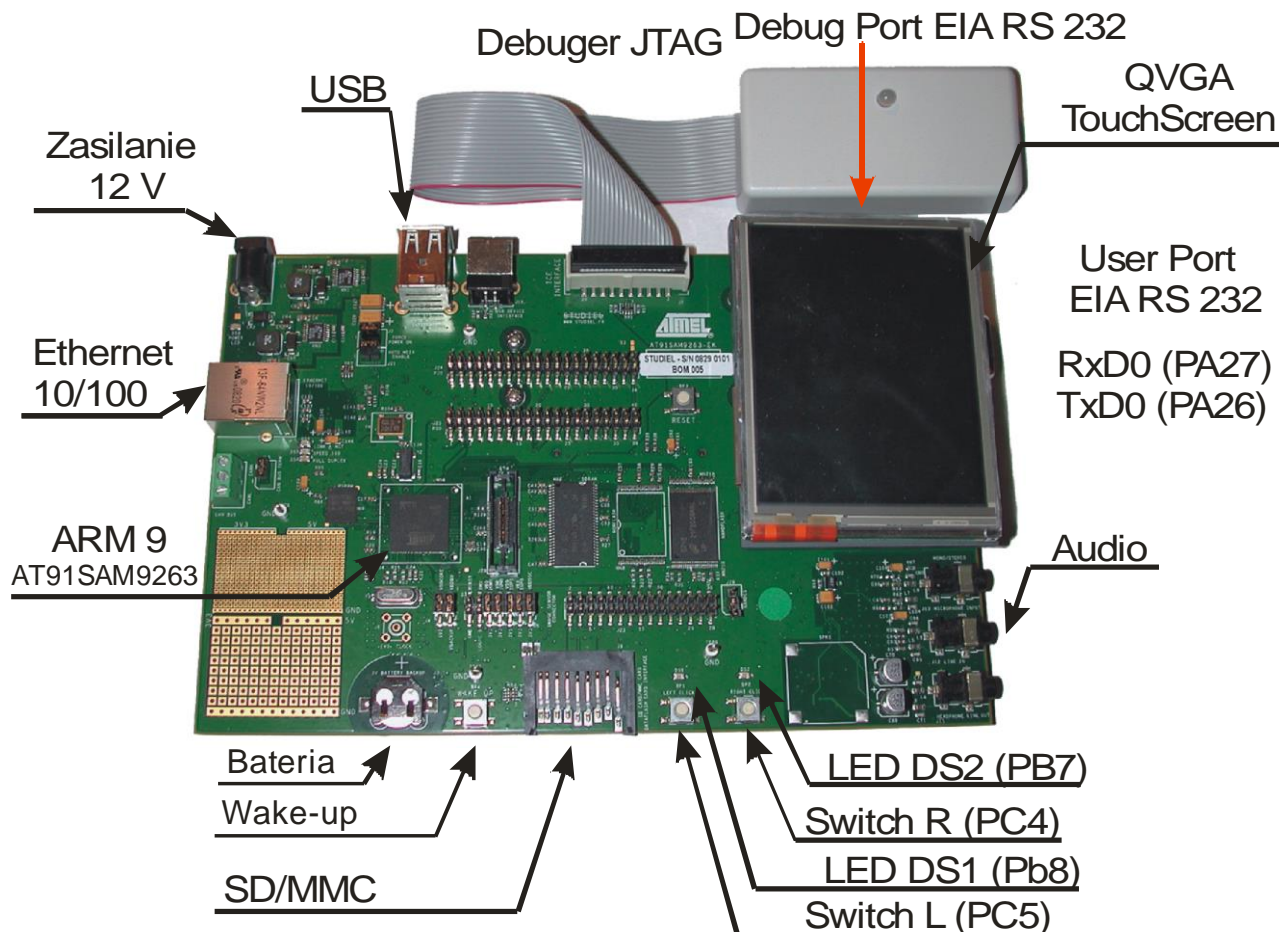
- pamięć ulotna SRAM – 80 kB pracująca z pełną szybkością procesora lub magistrali AMBA (0 waitstate),
- pamięć ulotna SRAM – 16 kB pracująca z pełną magistrali AMBA (0 waitstate),
- pamięć nieulotna FLASH – 128 kB pracująca z pełną szybkością magistrali AMBA (0 waitstate).



Procesor współpracuje z następującymi pamięciami zewnętrznymi:

- pamięć dynamiczna SDRAM – 256 MB (8 M x 32 bit, dostępna na szynie EBI0),
- pamięć statyczna PSRAM 4 MB (2 M x 16 bit, dostępna na szynie EBI1),
- pamięć nielotna NANDFLASH – 256 MB (256 M x 8, dostępna na szynie EBI0),
- pamięć nielotna NORFLASH 8 MB (4 M x 16, dostępna na szynie EBI0).

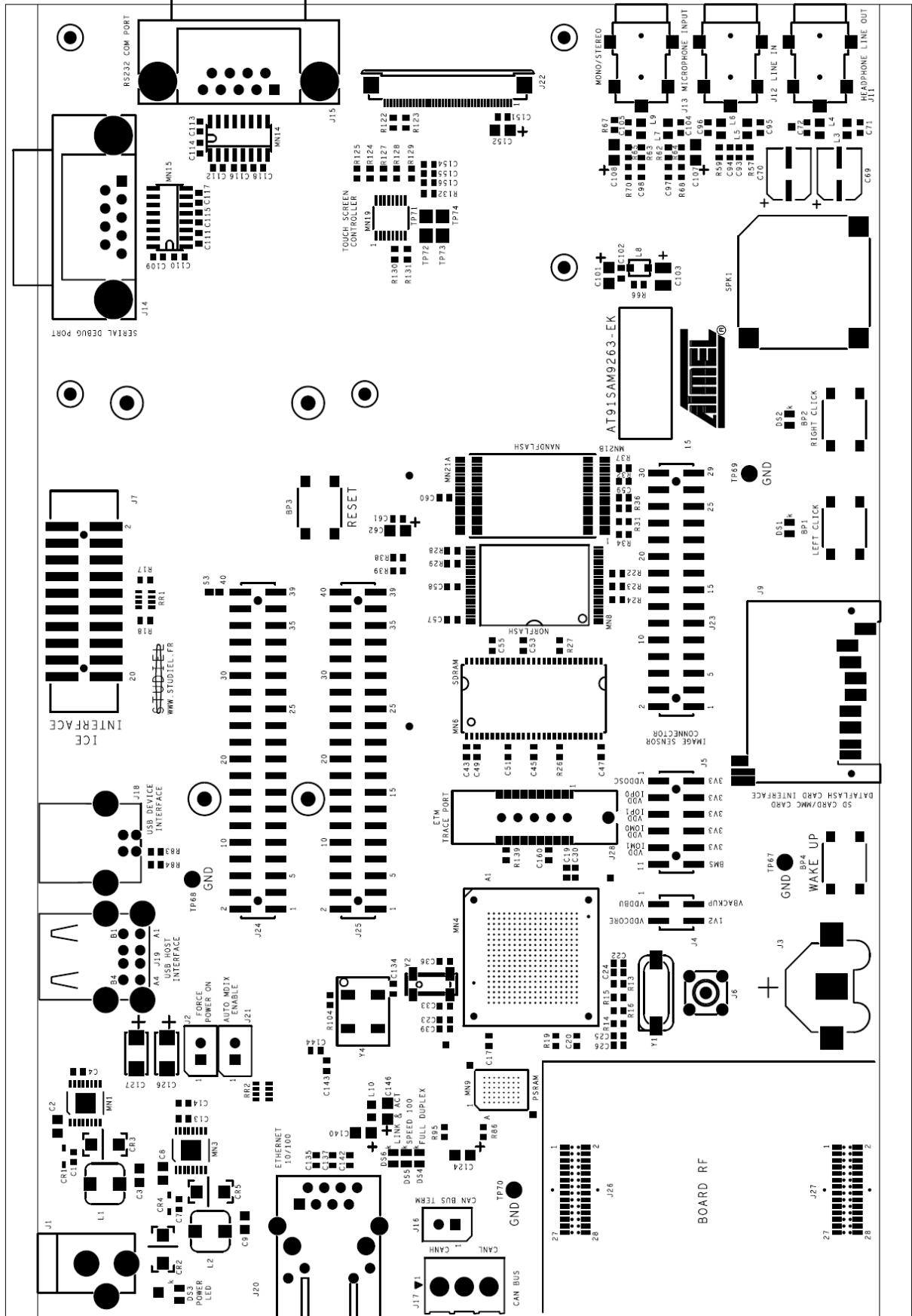
Na rysunku 2 zamieszczono zdjęcie płyty bazowej z zaznaczonymi elementami wykorzystywanymi do komunikacji z urządzeniami zewnętrznymi lub użytkownikiem systemu.



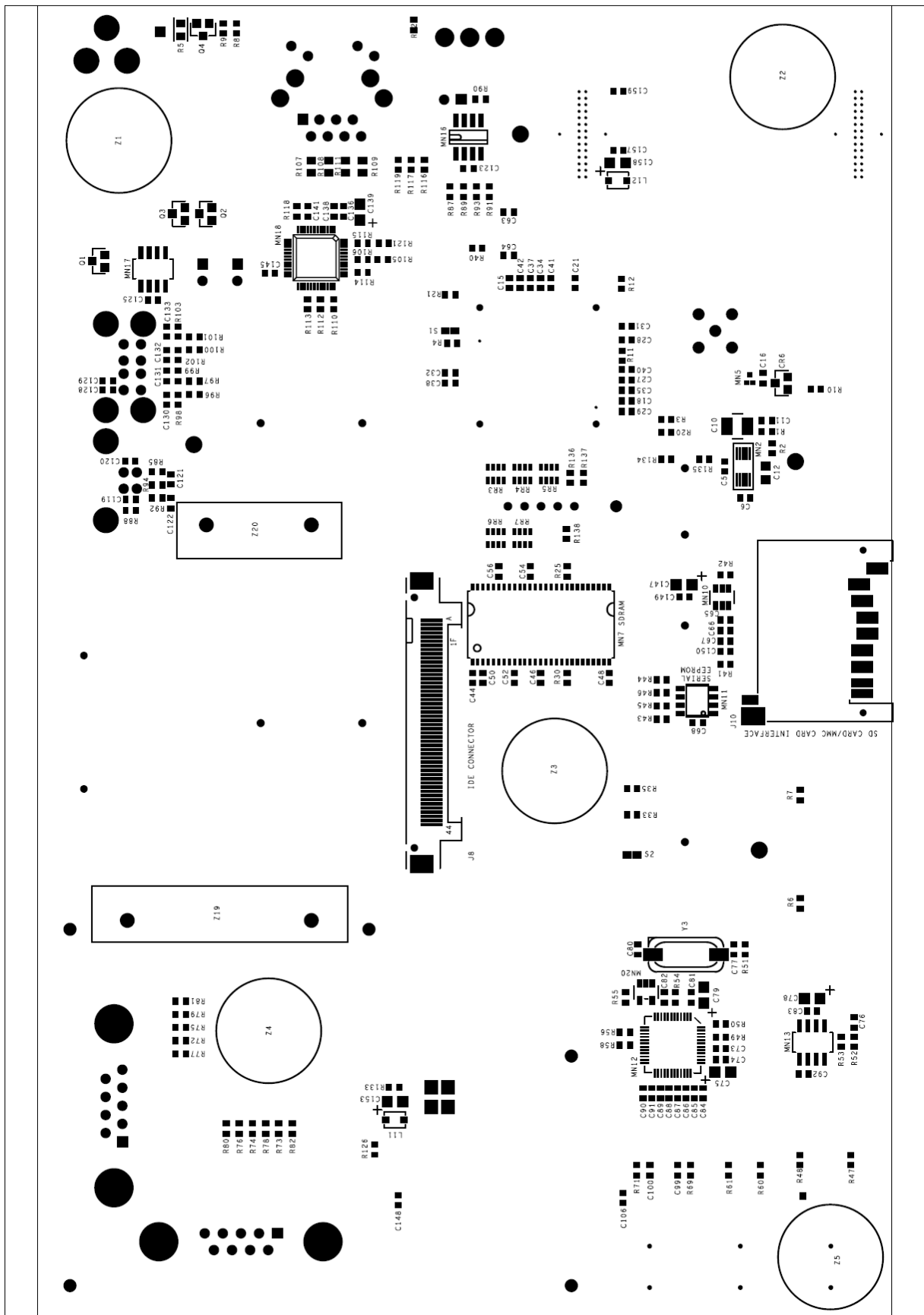
**Rys. 2. Płyta ewaluacyjna z procesorem AT91SAM9263**

Płyta wyposażona jest w szereg interfejsów oraz układów peryferyjnych:

- Dostępne interfejsy:
  - Ethernet 100-base TX,
  - USB FS device Master,
  - 2 x USB FS Host,
  - CAN 2.0B,
  - EIA RS232 (przeznaczone dla użytkownika),
  - EIA RS232 (przeznaczone dla celów diagnostycznych lub debugowania),
- Dostępne urządzenia peryferyjne:
  - Wyświetlacz: 3.5" 1/4 VGA TFT LCD z ekranem dotykowym,
  - Kodek audio: AC97 Audio,
  - Debug interface: JTAG,
  - Interfejs do programowania: JTAG,
  - Złącze kart SD/SDIO/MMC.



Rys. 3. Rozmieszczenie elementów na płytce bazowej – strona górna płytki



Rys. 4. Rozmieszczenie elementów na płytce bazowej – strona dolna płytki





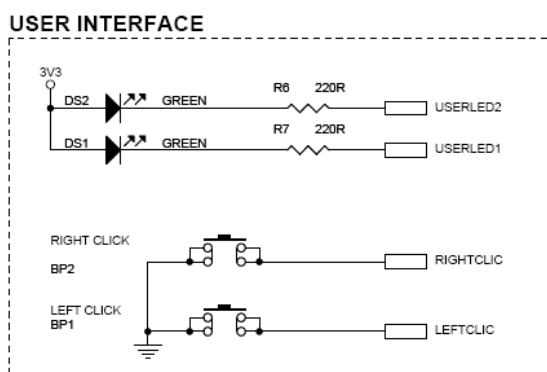
Na rysunku 3 i 4 przedstawiono rozmieszczenie elementów na warstwie górnej i dolnej płytki bazowej, natomiast na rysunku 5 przedstawiono schemat blokowy układu z zaznaczonymi interfejsami, układami pamięci i innymi urządzeniami peryferyjnymi.

### 3. Schematy połączeń urządzeń wykorzystywanych na laboratorium oraz rejestry sterujące

Na laboratorium wykorzystywane będą moduły portów I/O, transceiver UART, timery procesora, kontroler przerwań oraz urządzenia peryferyjne dostępne na płycie dodatkowej.

#### 3.1. Układy zamontowane na płycie bazowej

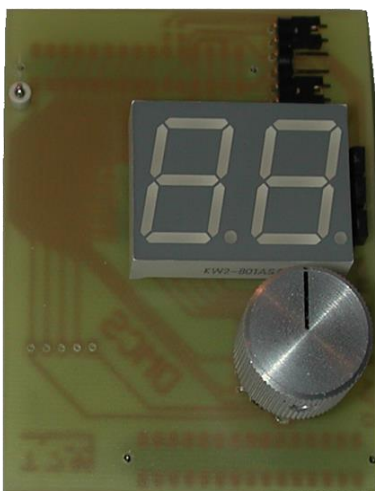
Na rysunku 4 przedstawiono schemat obrazujący dołączenie diod świecących LED oraz przełączników zwiernych push-button. Nazwy portów procesora, do których dołączone są diody LED i mikroprzełączniki przedstawiono na rys. 2 (opisy w nawiasach).



Rys. 4. Schemat obrazujący dostępne diody LED oraz mikroprzełączniki

#### 3.2. Układy zamontowane na płycie dodatkowej

Na rysunku 5 pokazano zdjęcie płytki dodatkowej (w wersji 1).



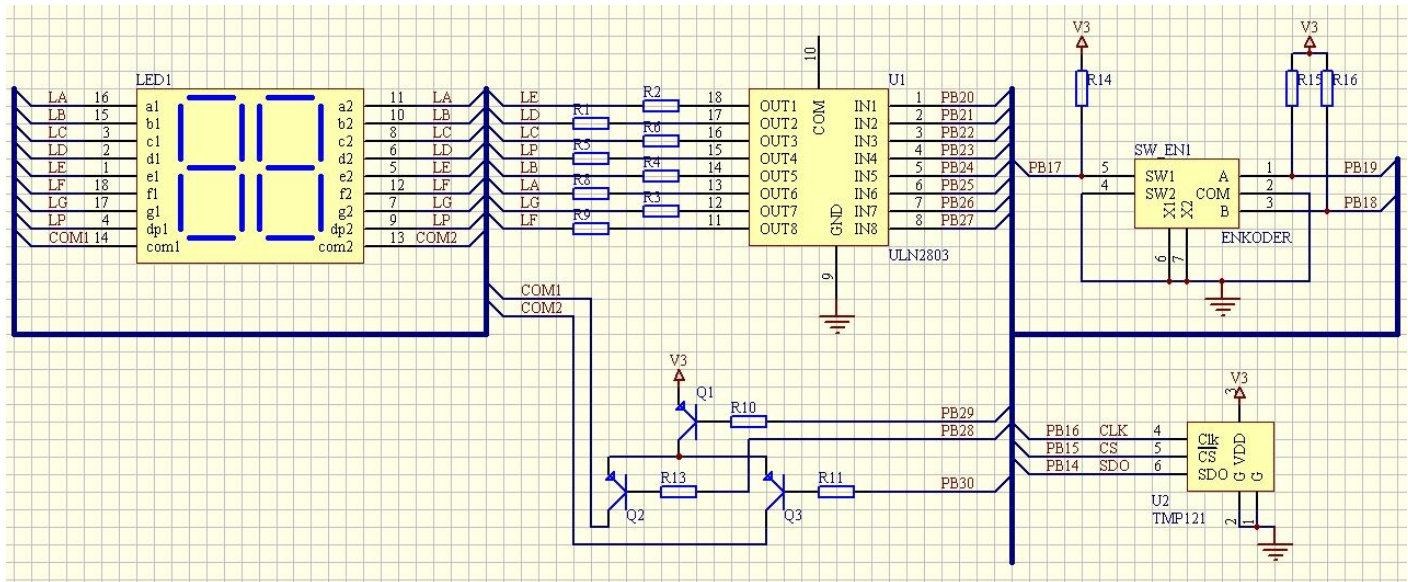
Rys. 5. Zdjęcie płytki z enkoderem i wyświetlaczem



Na płycie zainstalowane są następujące elementy:

- Enkoder obrotowy,
- Termometr cyfrowy,
- Wyświetlacz LED,
- Przetwornik dźwiękowy (buzzer).

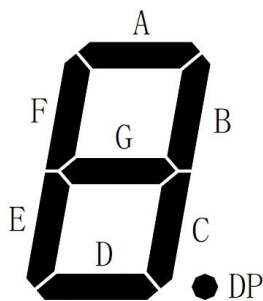
Schemat elektryczny układów pokazany jest na rysunku 6.



Rys. 6. Schemat płyty dodatkowej

### 3.2.1 Wyświetlacz LED

Na płycie znajduje się wyświetlacz podwójny 7 segmentowy. Oprócz wyświetlania cyfr i symboli można sterować kropką przy każdej cyfrze. Odpowiednie segmenty obu wyświetlaczy są połączone razem. Wyświetlacz jest sterowany dynamicznie, tylko jedna cyfra w danym momencie jest wyświetlana, wyboru cyfry dokonuje się przy pomocy sterowania tranzystorami Q2 i Q3. Tranzystory Q2 i Q3 załącza się ustawiając poziom wysoki odpowiednio na wyprowadzenia PB28 i PB30. Do celów testowych można oba albo jeden tranzystor załączyć na stałe. Każdy jeden segment można podświetlić odpowiednio ustawiając wyjście sterujące segmentami. Na rysunku 7 pokazano szkic pojedynczego segmentu z nazwanymi segmentami.



Rys. 7. Szkic wyświetlacza LED

| Nazwa segmentu wyświetlacza | Nazwa portu procesora |
|-----------------------------|-----------------------|
| Segment A                   | PB25                  |
| Segment B                   | PB24                  |
| Segment C                   | PB22                  |
| Segment D                   | PB21                  |
| Segment E                   | PB20                  |
| Segment F                   | PB27                  |
| Segment G                   | PB26                  |
| Kropka (DP)                 | PB23                  |

Istnieje możliwość sterowania poziomem jasności wyświetlacza przy pomocy PWM. Można to zrealizować programowo lub sprzętowo sterując tranzystorem Q1. Do testów można ten

tranzystor załączyć na stałe (pełna jasność). Tranzystor Q1 załącza się poziomem wysokim na wyprowadzeniach PB29.

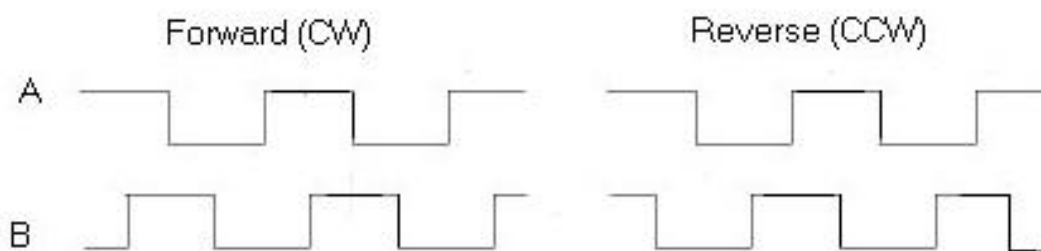
Aby wyświetlać na wyświetlaczu cyfry lub inne symbole należy w programie zaprojektować odpowiednią funkcję przekodowującą znak (cyfrę) na kombinację podświetlonych segmentów (np. przy pomocy tablicy). Aby wykorzystać obie cyfry wyświetlacza należy odpowiednio szybko (min. 30 Hz) wyświetlać obie cyfry naprzemiennie.

### 3.2.2 Enkoder obrotowy

Enkoder obrotowy zainstalowany na płycie składa się z dwóch niezależnych elementów, przycisku monostabilnego i właściwego dwukanałowego enkodera kwadraturowego.

Wciśnięcie gałki powoduje uaktywnienie przycisku, natomiast obrót gałką powoduje wygenerowanie serii impulsów. Wciśnięcie może być wykrywane niezależnie od jednoczesnego obracania gałką. Aktywne wciśnięcie jest sygnalizowane stanem niskim na wyjściu enkodera.

W zależności od kierunku obrotów impulsy z enkodera mają postać przedstawioną na przebiegach czasowych na rysunku 8.



Rys. 7. Przebiegi czasowe na wyjściu enkodera

Aby rozpoznać kierunek obrotów w najprostszym przypadku należy uznać jeden z kanałów jako kanał referencyjny i wybrać arbitralnie jedno ze zboczy a następnie zawsze na tym zboczu sprawdzać stan logiczny na drugim kanale. Enkoder jest podłączony w następujący sposób do procesora:

- Kanał A: PB19
- Kanał B: PB18
- Przycisk: PB17

Oprócz wykrywania kierunku obrotów można mierzyć szybkość obracania gałką, w tym celu należy mierzyć czas pomiędzy impulsami (lub częstotliwość) jednego z kanałów.

### 3.2.3 Termometr

Zainstalowany jest termometr typu TMP121 lub TMP123. Specyfikacja układu dostępna jest w postaci wydrukowanej w laboratorium lub na stronie producenta ([www.ti.com](http://www.ti.com)). Obsługa termometru polega na programowym generowaniu przebiegów sygnałów według specyfikacji. Termometr komunikuje się przy pomocy interfejsu SPI (Serial Peripheral Interface). Wyprowadzenia układu są podłączone następująco:

- Zegar (CLK, SPI1\_SPCK): PB14
- Dane (SDO/MISO, SPI1\_MISO): PB12
- Sygnał wyboru układu (CS, SPI1\_NPCS0): PB15

## 4. Korzystanie z oprogramowania dostępnego w laboratorium

Na zajęciach laboratoryjnych wykorzystywane są następujące narzędzia dostępne na licencji GNU (<http://www.gnuarm.org/>):

- GCC-4.3 toolchain (Linux):
  - Kompilator: gcc-4.3.2,
  - Przydatne narzędzia: binutils-2.19,
  - Biblioteki C/C++: newlib-1.16,
  - Debugger zgodny z GDB: insight-6.8.

### 4.1 Procesory ARM

- Asemblacja programów napisanych w języku asemblera
  - Przejść do katalogu, w którym znajduje się program źródłowy
  - Wpisać polecenie `arm-elf-asm <nazwa_pliku>`, aby zasemblować program. Jeżeli asemblacja się powiedzie, zostanie wygenerowany plik z rozszerzeniem elf i takiej samej nazwie jak nazwa pliku źródłowego
  - Można skorzystać z programu `make`, ale wtedy lepiej bezpośrednio wykorzystać do asemblacji program **arm-elf-asm**
- Kompilacja programów w języku C
  - Przejść do katalogu, w którym znajduje się program źródłowy
  - Wpisać polecenie `arm-elf-gcc <nazwa_pliku>`
  - Można skorzystać z programu `make`, ale wtedy lepiej bezpośrednio wykorzystać do asemblacji program **arm-elf-gcc**

### 4.2 Program debugera GDB

- Program debugera uruchamia się poleceniem: `arm-elf-gdb`
- **W przypadku procesorów ARM konieczne jest wcześniejsze uruchomienie programu OpenOCD i utrzymywanie tego programu uruchomionego przez cały czas (np. na innej konsoli)**
- Ustawianie pułapek, polecenie breakpoint (w skrócie b) `<numer linii>` lub `<nazwa funkcji>`
- Uruchamianie programu, polecenie run (w skrócie r)
- Po wejściu w pułapkę, można kontynuować wykonanie programu poleceniem continue (c), lub wykonywać go krok po kroku z wchodzeniem do wnętrza funkcji - polecenie step (s) lub bez wchodzenia - polecenie next (n)
- Aby powrócić do wykonywania funkcji nadrzędnej należy wpisać polecenie finish
- Aby wyświetlić zawartość rejestru lub danej należy skorzystać z polecenia print (p)
- Aby przerwać działanie programu należy wcisnąć kombinację CTRL+C
- Aby zakończyć pracę z debugerem należy wpisać polecenie quit (q)

### 4.3 Program minicom

- Uruchomić program Minicom poleceniem *minicom*
- W przypadku jeśli program **minicom** nie daje się uruchomić i kończy się z błędem braku

- możliwości zapisania pliku „lock-file”, należy uruchamiać program **minicom** z parametrem -t
- Po zainicjowaniu komunikacji, znaki odebrane z mikrokontrolera będą wyświetlane w oknie terminala
  - Aby wysłać znak należy nacisnąć odpowiedni klawisz na klawiaturze PC
  - Aby wyczyścić zawartość okna terminala należy wybrać kombinację **CTRL+A C**
  - Aby opuścić program **minicom** należy wybrać kombinację **CTRL+A Q**
  - Aby zmienić parametry programu **minicom** należy wybrać kombinację CTRL+A O
  - Przy pierwszym uruchomieniu programu należy skonfigurować przynajmniej:
    - Urządzenie portu szeregowego /dev/ttyS0 lub /dev/ttyS1, jeżeli urządzenie nie jest znane proszę wykorzystać polecenie **dmesg / grep tty**
    - Prędkość, parzystość i ilość bitów (wedle własnego programu)
    - Proszę wyłączyć kontrolę przepływu danych, zarówno sprzętową, jak i programową
  - Zaleca się zapisać domyślną konfigurację programu, opcja ‘save as dfl’
  - Jeśli w czasie działania programu zmienia się urządzenie portu szeregowego, należy zapisać konfigurację, zakończyć program i uruchomić go ponownie