

Instrukcja do laboratorium SMCR oraz PM

I. *Regulamin pracy w laboratorium*

1. Zaliczanie laboratorium

- Obecność na zajęciach jest obowiązkowa. Dopuszczalna jest jedna nieusprawiedliwiona nieobecność w ciągu semestru.
- Studenci wykonują zaplanowane ćwiczenia w grupach dwuosobowych.
- Każde ćwiczenie jest indywidualnie oceniane, z uwzględnieniem jakości i terminowości jego wykonania. Ocena i ew. uwagi są zapisywane w dzienniku.
- Na pierwszych zajęciach każda grupa otrzymuje dziennik pracy w laboratorium. Jego okazanie jest warunkiem koniecznym do przystąpienia do zajęć.
- W przypadku stwierdzenia nieprzygotowania do zajęć, student jest z nich usuwany. Jest to równoznaczne z nieobecnością nieusprawiedliwioną.
- Ocena końcowa jest średnią ocen otrzymanych za poszczególne ćwiczenia.
- Rażące spóźnienie na zajęcia powoduje niedopuszczenie do ich odbycia.

2. Kategorycznie zabrania się:

- Zdejmowania obudów komputerów;
- Podłączania i odłączania jakichkolwiek przewodów bez zezwolenia osoby prowadzącej zajęcia;
- Dotykania zestawów uruchomieniowych w celu innym niż naciśnięcie przycisku RESET;
- Wnoszenia jakichkolwiek przedmiotów (w tym dokumentacji technicznej) znajdujących się w laboratorium;
- Przepisywania programów od innych grup;
- Przebywania w laboratorium osobom nie odbywającym zajęć i nie związanych bezpośrednio z obsługą laboratorium.

3. W przypadku naruszenia zakazów wymienionych w pkt. 2, 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.

II. *Opis sprzętu znajdującego się w laboratorium*

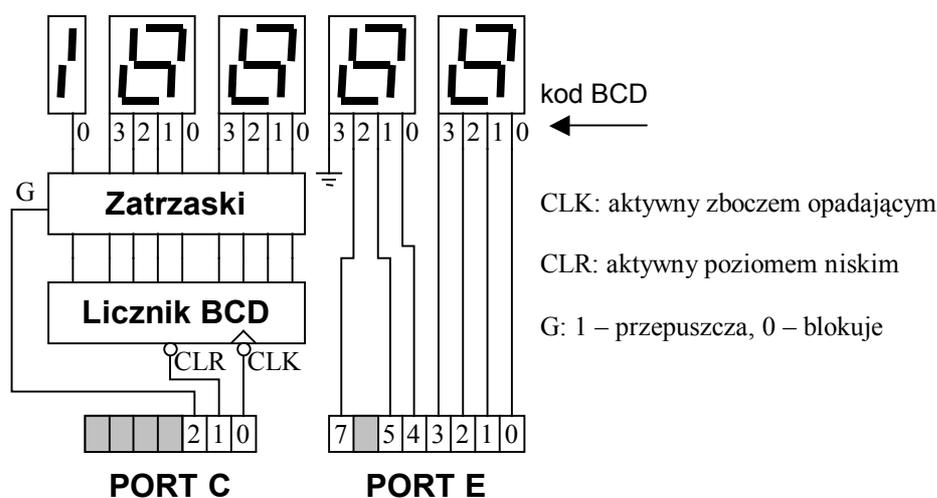
W laboratorium znajduje się następujący sprzęt:

1. Komputery klasy PC

2. Zestawy uruchomieniowe firmy ELKOMTECH, oparte na 32-bitowym mikrokontrolerze Motorola MC68331, dysponujące następującymi zasobami:

- 32-bitowy mikroprocesor oparty na jądrze CPU32, kompatybilnym z rodziną procesorów 68020
- 512 kB pamięci FLASH przeznaczonej na program, z nagrany bootloaderem w sektorze nr 0; adresy: 0x000000 – 0x07FFFE
- 256 kB pamięci statycznej RAM, przeznaczonej na program i dane; adresy: 0x100000 – 0x13FFFE

- kanał BDM (*Background Debug Mode*) umożliwiający załadowanie nowego programu do mikrokontrolera oraz kontrolowanie jego działania
 - port szeregowy asynchroniczny zewnętrznie kompatybilny z RS-232, ze złączem DB9, a wewnętrznie podłączony do wbudowanego w mikrokontroler asynchronicznego interfejsu szeregowego SCI (*Serial Communication Interface*)
 - interfejs szeregowy synchroniczny QSPI (*Queued Serial Communication Interface*) (odpowiednik magistrali I²C)
 - 7-bitowy równoległy port wyjściowy C, multipleksowany z sygnałami CS (*Chip Select*); dostępne są końcówki PC₂..PC₀
 - 8-bitowy dwukierunkowy port równoległy E, multipleksowany z sygnałami sterującymi magistralami; dostępne są końcówki PE₇;PE₅..PE₀
 - 8-bitowy dwukierunkowy port równoległy F, multipleksowany z liniami, którymi są zgłaszane przerwy zewnętrzne; dostępne są końcówki PF₇..PF₁
 - 8-bitowy dwukierunkowy port równoległy G, multipleksowany z końcówkami IC (*Input Capture*) lub OC (*Output Compare*) timera; dostępne są końcówki PG₇..PG₀
 - Wyjście PWM (*Pulse Width Modulation*)
3. Przenośne zasilacze stabilizowane 220 V ~/9V DC, zapewniające prąd o natężeniu do 700 mA
 4. Adaptery łączące kanały BDM mikrokontrolerów z komputerami PC przez jego port równoległy (LPT)
 5. Cyfrowe panele odczytowe LED, z osobnym zasilaniem, podłączone do zestawów uruchomieniowych przez porty C i E w sposób pokazany na rysunku



UWAGA! Końcówek portów zaznaczonych szarym kolorem nie wolno używać, gdyż są one multipleksowane z ważnymi sygnałami sterującymi!

6. Przewody łączące zestawy uruchomieniowe z komputerami PC przez kanał szeregowy (COM)
7. Klawiatury 4x4 podłączone do zestawów uruchomieniowych przez port G
8. Wspólny zasilacz dla wszystkich paneli odczytowych 220 V ~/5V DC, zapewniający prąd o maks. natężeniu 20 A

III. Korzystanie oprogramowania dostępnego w laboratorium

1. Asemblacja i uruchamianie programów napisanych w języku asemblera
 - Przejść do katalogu, w którym znajduje się program źródłowy
 - Wpisać polecenie **asm-cpu32 <nazwa programu>** aby zasemblować program; jeżeli asemblacja się powiedzie, zostanie wygenerowany plik z rozszerzeniem **.bin**
 - Uruchomić debugger skrośny poleceniem **gdb-bdm <program.bin>** z konsoli lub ew. **gdb-bdm-gui <program.bin>** w środowisku graficznym X
 - Załadować program poleceniem **load** z konsoli lub z menu graficznego (urządzenie **BDM ICD32, /dev/icd_bdm0**)
 - Zastawić ew. pułapki poleceniem **breakpoint (b)**
 - Uruchomić program poleceniem **run (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 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 debuggerem należy wpisać polecenie **quit (q)**
2. Kompilacja i uruchamianie programów w języku C, przeznaczonych dla komputera PC
 - Przejść do katalogu, w którym znajduje się program źródłowy
 - Aby go skompilować, wpisać polecenie
gcc -g -o nazwa_programu_wynikowego programy_zrodlowe
 - Uruchomić debugger lokalny poleceniem **gdb -nw <program_wynikowy>** z konsoli lub ew. **gdb <program_wynikowy>** w środowisku graficznym X i postępować dalej podobnie jak w pkt. 1, pomijając fazę ładowania programu
3. Kompilacja i uruchamianie programów napisanych na mikrokontroler Motorola 68331, nie korzystających z bibliotek standardowych języka C
 - Przejść do katalogu, w którym znajduje się program źródłowy i skopiować do niego plik **/usr/local/m68k-elf/templates/nolibs/makefile**
 - W pliku **makefile** zmienić **example.bin** na **<program_zrodlowy>.bin**
 - Uruchomić kompilację skrośną poleceniem **make** i obserwować komunikaty kompilatora i konsolidatora
 - Uruchomić debugger skrośny i korzystać z niego w analogiczny sposób jak w pkt. 1
4. Kompilacja i uruchamianie programów napisanych na mikrokontroler Motorola 68331, korzystających z bibliotek standardowych języka C (włącznie z **lib331.a**)
 - Przejść do katalogu, w którym znajduje się program źródłowy i skopiować do niego plik **/usr/local/m68k-elf/templates/c++/makefile**
 - W pliku **makefile** zmienić **example.bin** na **<program_zrodlowy>.bin**
 - Uruchomić kompilację skrośną poleceniem **make** i obserwować komunikaty kompilatora i konsolidatora
 - Uruchomić debugger skrośny i korzystać z niego w analogiczny sposób jak w pkt. 1
5. Odbieranie i wysyłanie znaków z i do mikrokontrolera na komputerze PC
 - Na drugiej konsoli uruchomić program **minicom**

- Po zainicjalizowaniu 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**
6. Biblioteki i pliki nagłówkowe specyficzne dla sprzętu znajdującego się w laboratorium
- Podstawową biblioteką napisaną specjalnie dla zestawu uruchomieniowego znajdującego się w laboratorium jest biblioteka **331 (lib331.a)**. Zapewnia ona następujące funkcje:
- Przekierowuje wejście/wyjście standardowe na port szeregowy. Dzięki temu można wysyłać i odbierać znaki stosując zwykle funkcje z biblioteki **stdio**.
 - Odbierane znaki są buforowane. Bufor wejściowy mieści 63 znaki.
 - Umożliwia dynamiczną alokację pamięci na sterce przy pomocy funkcji **malloc**, **free** i **realloc**
 - Przenosi tablicę wektorów przerwań i wyjątków z pamięci FLASH do pamięci RAM, umożliwiając tym samym jej modyfikację przez program. Położenie tej tablicy w pamięci RAM określa 256-elementowa tablica **exception_vectors[]**
 - Uaktywnia obsługę przerwań zgłoszonych na poziomie 3 lub wyższym

Pliki nagłówkowe specyficzne dla mikrokontrolera 68331:

- *lib331.h* - Prototypy funkcji, zmienne globalne i typy danych zdefiniowane w bibliotece 331
- *sim.h* - Mapa pamięci i wartości masek dla modułu SIM
- *gpt.h* - Mapa pamięci i wartości masek dla modułu GPT (timer)
- *qsm.h* - Mapa pamięci i wartości masek dla modułu QSM (SCI, SPI)