Instrukcja do laboratorium SMCR oraz PM

Freescale MCF5282

1. Regulamin pracy w laboratorium

a) Zaliczenie 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.
   - 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.

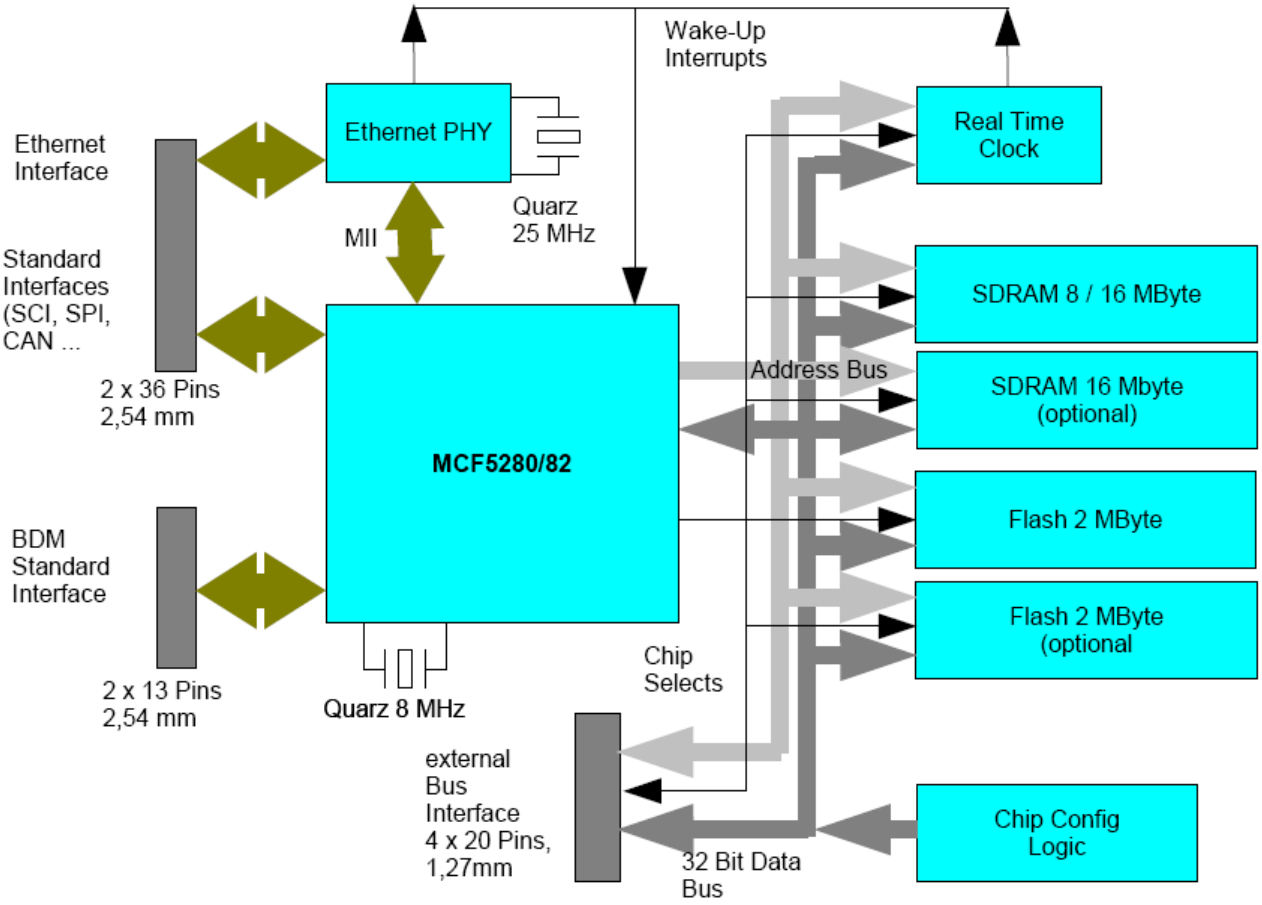
b) Kategorycznie zabrania się:
   - Zdejmowania obudów komputerów.
   - Podłączania i odłączania jakichkolwiek przewodów bez zezwolenia osoby prowadzącej zajęcia.
   - Dotykania zestawów uruchomieniowych w celu innym niż naciśnięcie przycisku RESET lub IRQ7.
   - Wynoszenia jakichkolwiek przedmiotów (w tym dokumentacji technicznej) znajdujących się w laboratorium.
   - Przepisywania lub kopiowania programów, procedur lub funkcji od innych grup.
   - Przebywania w laboratorium osobom nie odbywającym zajęć i nie związanych bezpośrednio z obsługą laboratorium.

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 oparte na 32-bitowym mikrokontrolerze z rdzeniem Coldfire 2 MFC5282.
3. Płytka rozszerzająca z wyświetlaczem LCD, termometrem SPI oraz potencjometrem.
4. Przenośne stabilizowane zasilacze 5V/1A.
5. Adaptery łączące kanał diagnostyczny BDM mikrokontrolerów z komputerem PC.
6. Cyfrowe panele odczytowe LED.
7. Przewody łączące zestawy uruchomieniowe z komputerami PC przy użyciu interfejsu EIA 232.
8. Klawiatury numeryczne 3x4.
9. Wspólny zasilacz o napięciu 5 V i wydajności prądowej 20 A.



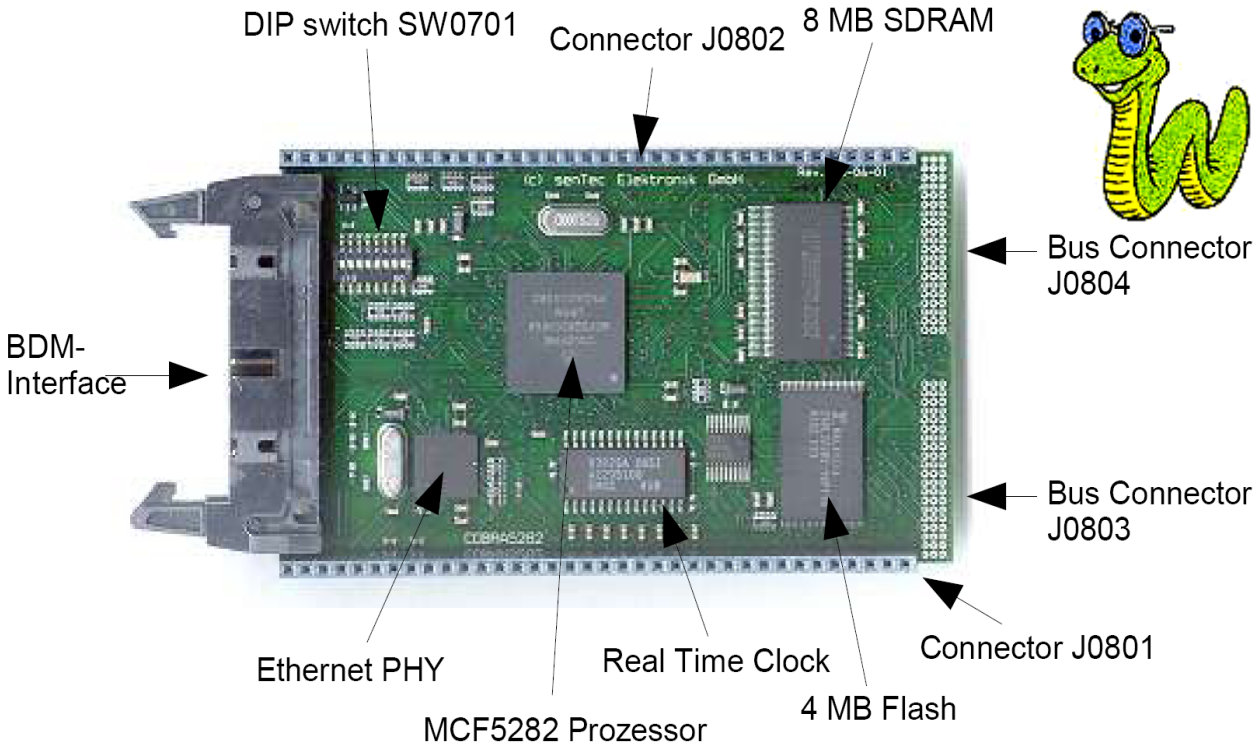
Schemat blokowy modułu COBRA 5282

Zestawy uruchomieniowe składają się z trzech modułów:

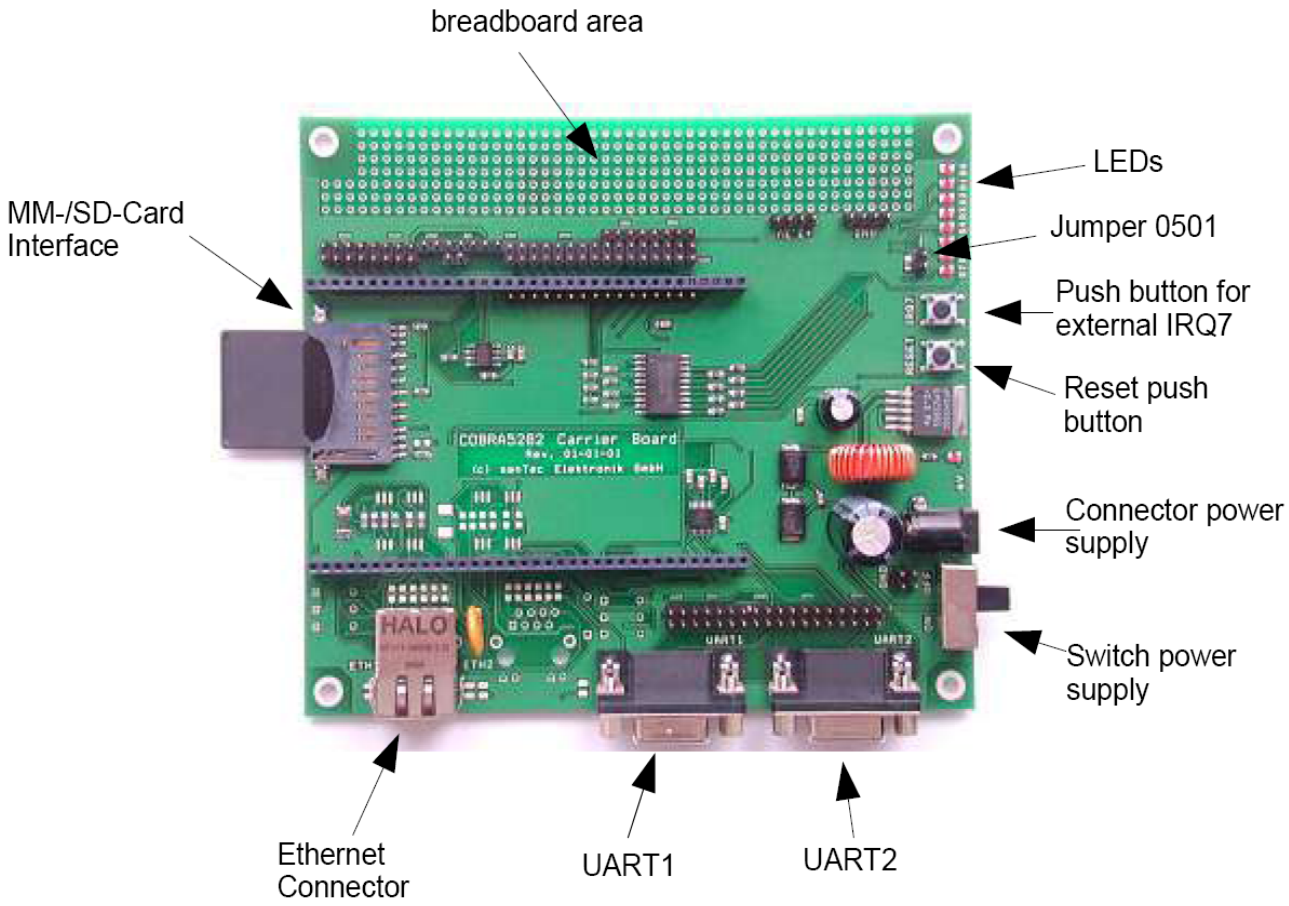
1. Płyty bazowej
2. Modułu COBRA 5282
3. Modułu rozszerzającego

Moduł COBRA 5282 zbudowany jest w oparciu o mikrokontroler Freescale MCF5282. Procesor współpracuje z 2 MB zewnętrzną pamięcią FLASH, w której przechowywany jest monitor dBUG

ROM oraz 8 MB pamięci SDRAM wykorzystywanej do przechowywania głównego programu. Program ładowany jest i uruchamiany przy pomocy kanału diagnostycznego BDM poprzez interfejs BDM dołączony do portu równoległego lub USB komputera PC. Interfejs BDM ułatwia debugowanie załadowanego programu.
Moduł wyposażony jest dodatkowo w zegar czasu rzeczywistego V3025. Komunikacja z procesorem możliwa jest przy użyciu interfejsu Ethernet 10/100 Mbit oraz dwu programowalnych transmiterów UART kompatybilnych ze standardem EIA 232. Na płycie bazowej umieszczono złącze pamięci SD/MMC. Mikrokontroler MCF5282 został wyposażony w dużą liczbę urządzeń peryferyjnych. Na laboratorium wykorzystywane będą moduły portów I/O, transmiter UART, timery procesora, kontroler przerwań, przetwornik ADC oraz transmiter SPI.

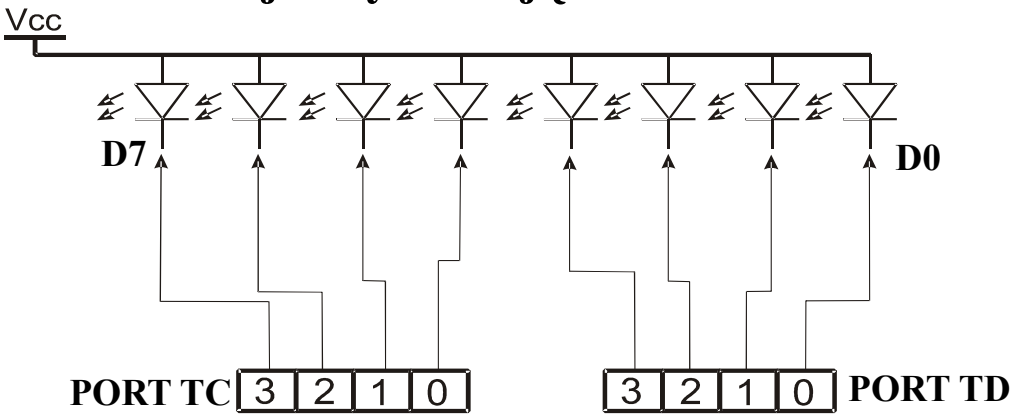


Moduł COBRA 5282

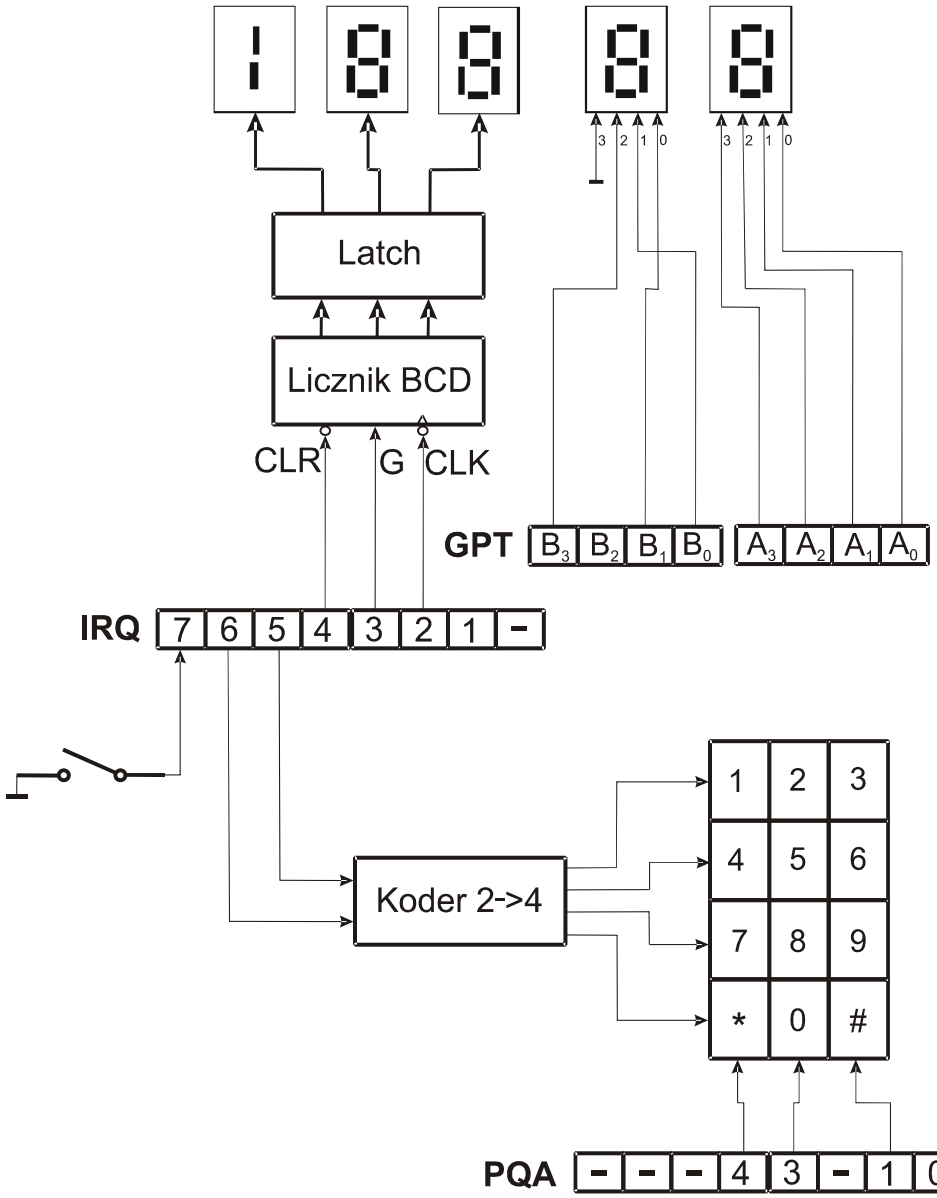


Płytka bazowa zestawu uruchomieniowego wykorzystywanego na laboratorium

3. Schematy połączeń urządzeń wykorzystywanych na laboratorium oraz rejestry sterujące

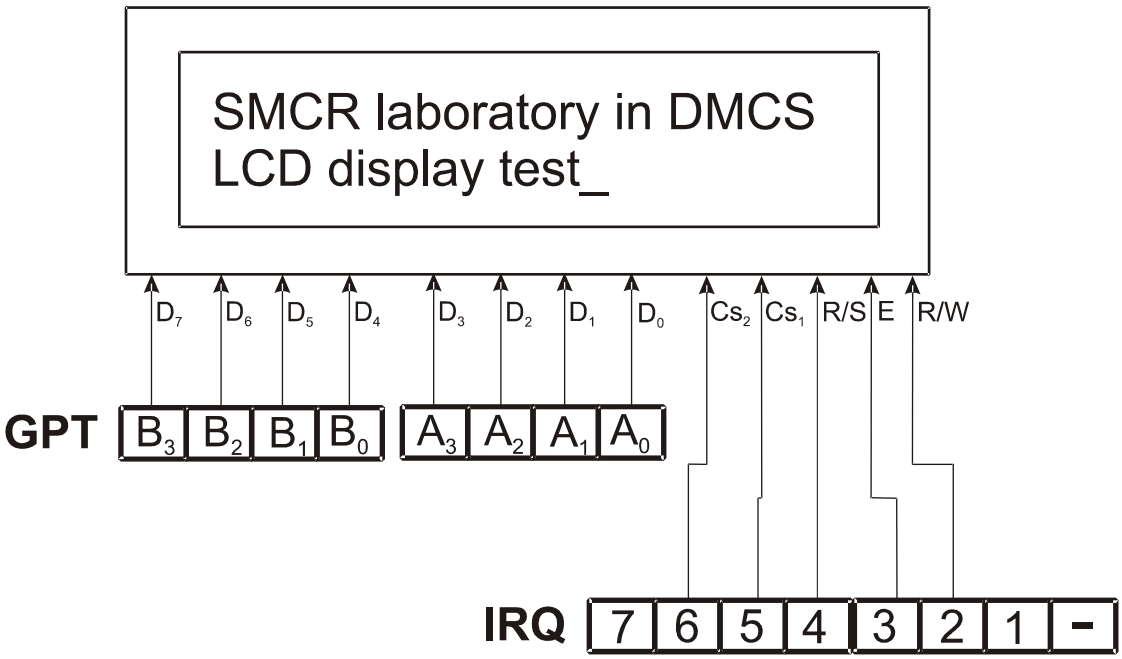


Podłączenie diod LED

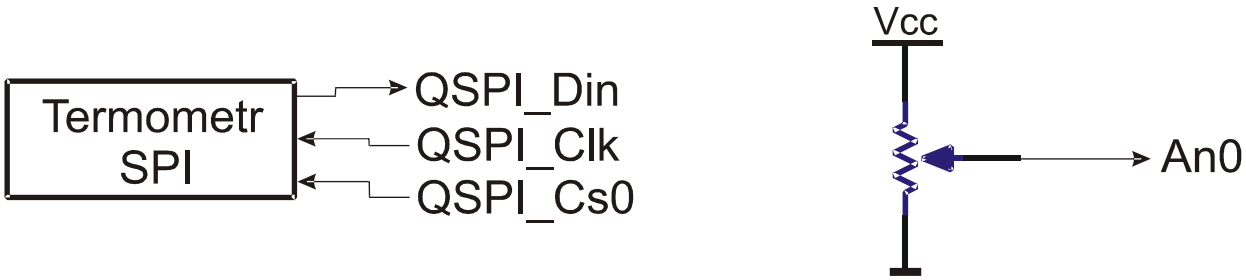


Podłączenie wyświetlacza LED oraz klawiatury matrycowej

(podczas obsługi dwóch młodszych cyfr wyświetlacza LED bit IRQ 5 musi być ustawiony na '0')



Podłączenie wyświetlacza LCD
(bity sterujące CS1, CS2 tylko dla wyświetlacza matrycowego)



Podłączenie termometru TMP 123 oraz potencjometru

Przestrzeń adresowa	*Sygnał sterujący*	*Urządzenie*
$0000 0000 - $04FF FFFF	SDRAM space	Pamięć SDRAM 16 MB
$1000 0000 - $1000 FFFF	~CS2	Zegar czasu rzeczywistego
$2000 0000 - $2000 FFFF	internal SRAM	Wewnętrzna pamięć SRAM 4 kB
$4000 0000 - $4000 FFFF		Rejestry procesora
$F000 0000 - $F007 FFFF	internal FLASH	Wewnętrzna pamięć FLASH 512 kB
$FFC0 0000 - $FFFF FFFF	~CS0	Pamięć FLASH 2 MB

Mapa pamięci modułu COBRA

4. Korzystanie z oprogramowania dostępnego na laboratorium z przedmiotu PM

a) Kompilacja i uruchamianie programów napisanych w języku C, przeznaczonych dla komputera PC
   - Przejść do katalogu, w którym znajduje się program źródłowy
   - Aby go skompilować, wpisać polecenie:
     gcc -g -Wall -pedantic <program_źródłowy> -o <program_wynikowy>
     np.:
     gcc -g -Wall -pedantic hello.c -o hello.out
   - Uruchomić program wpisując jego nazwę w linii poleceń, np:
     hello.out

b) Kompilacja i uruchamianie programów napisanych w języku C, przeznaczonych dla mikrokontrolera Freescale MCF5282
   - Przejść do katalogu, w którym znajduje się program źródłowy. **Uwaga!!!** Nazwa katalogu nie może zawierać spacji
   - Aby skompilować program o nazwie *hello.c*, wpisać polecenie:
     /opt/gnutools/bin/m68k-elf-gcc -c -gdwarf-2 -g3 -W -Wall -m528x -O2 -save-temps -I/opt/labtools/pm hello.c -o hello.o
   - Aby go zlinkować, wpisać polecenie:
     /opt/gnutools/bin/m68k-elf-gcc -o hello.out -gdwarf-2 -g3 -m528x -L/opt/labtools/pm hello.o -Wl,-Map,hello.map -Wl,-T,/opt/labtools/pm/lscript528x.ld
   - Uruchomić debuger skrośny poleceniem:
     gdb-cf hello.out
   - W środowisku debugera uruchomić program poleceniem *run* (wystarczy wpisać ***r***)
   - Aby przerwać działanie programu należy wcisnąć kombinację *CTRL+C*
   - Aby zakończyć pracę z debugerem należy wpisać polecenie *quit **(q)***

c) Kompilacja i uruchamianie programów napisanych w języku C, przeznaczonych dla mikrokontrolera Freescale MCF5282, z wykorzystaniem programu make.
   - Przejść do katalogu, w którym znajduje się program źródłowy i skopiować do niego plik */opt/labtools/pm/makefile*
   - W pliku makefile zmienić nazwę pliku wynikowego *hello.out* odpowiednio do nazwy własnego programu
   - Uruchomić kompilację skrośną poleceniem *make* i obserwować komunikaty kompilatora i konsolidatora
   - Uruchomić debuger skrośny poleceniem *make run*
   - W środowisku debugera uruchomić program poleceniem *run* (***r***)
   - Aby przerwać działanie programu należy wcisnąć kombinację *CTRL+C*
   - Aby zakończyć pracę z debugerem należy wpisać polecenie *quit **(q)***
   - Polecenie *make* generuje pliki pośrednie. Aby je usunąć należy wpisać *make clean*

d) Odbieranie i wysyłanie znaków z i do mikrokontrolera na komputerze PC

- Na drugiej konsoli uruchomić program minicom wpisując *minicom* w linii poleceń
- 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 zmienić parametry programu minicom należy wybrać kombinację *CTRL+A O*. Typowe ustawienia najważniejszych parametrów minicoma to:
  - *Urzadzenie szeregowe : /dev/ttyS0*
  - *Bps/Parzystosc/Bity : 19200 8N1*
  - *Sprzetowa kontrola przeplywu : No*
  - *Programowa kontrola przeplywu : No*

  Uwaga!!! Zmiana niektórych ustawień (np. */dev/ttyS0*) wymaga ponownego uruchomienia programu minicom
- Aby zapisać ustawienia wybieramy opcję "Zapisz setup jako dfl"
- Aby opuścić program minicom należy wybrać kombinację *CTRL+A Q*

e) Biblioteka specyficzna dla sprzętu znajdującego się w laboratorium

Podstawową biblioteką napisaną specjalnie dla zestawu uruchomieniowego znajdującego się w laboratorium jest biblioteka 528x (*lib528x.a*). Prototypy funkcji, zmienne globalne i typy danych zdefiniowane w bibliotece znajdują się w pliku nagłówkowym *mcf5282.h*. Biblioteka zapewnia następujące funkcje:

- Przekierowuje wejście/wyjście standardowe na port szeregowy. Dzięki temu można wysyłać i odbierać znaki stosując zwykłe funkcje z biblioteki stdio
- Odbierane znaki są buforowane. Bufor wejściowy mieści 63 znaki
- Umożliwia dynamiczną alokację pamięci na stercie przy pomocy funkcji malloc, free, i realloc
- Uaktywnia obsługę przerwań zgłoszonych na poziomie 3 lub wyższym

f) Debugowanie programów napisanych w języku C, przeznaczonych na komputer PC

- Aby możliwe było debugowanie programu przy pomocy debugera GDB należy skompilować program z użyciem opcji ***-g***:
  *gcc **-g** -Wall -pedantic program.c -o program.out*
- Aby rozpocząć pracę debugera należy z poziomu katalogu z plikiem wynikowym wywołać polecenie:
  gdb program.out
- Po uruchomieniu debugera możliwe jest wyświetlenie listingu programu za pomocą komendy *list (**l**)*.
  (gdb) l <opcjonalny_nr_linii lub nazwa_funkcji>
  np.:
  (gdb) l 4
  (gdb) l main
- W celu ustawienia pułapki (ang. breakpoint) w określonej linii programu (na początku funkcji lub pod pewnym adresem) używamy komendy *break (**b**)*, np:
  (gdb) b 5 - ustawia pułapkę w linii 5 programu
  (gdb) b main - ustawia pułapkę na początku funkcji main
- Program uruchamiamy poleceniem *run* (***r***)
  (gdb) r
- Program wykonywany jest do linii, w której znajduje się pułapka. Wykonanie programu

można kontynuować poleceniem *continue (**c**)*, lub wykonać go krok po kroku z wchodzeniem do wnętrza funkcji – polecenie *step (**s**)*, lub bez wchodzenia – polecenie *next (**n**)*:

(gdb) n
(gdb) s
(gdb) c

- Przy każdym kolejnym kroku możliwe jest wyświetlanie aktualnej wartości zmiennej jednorazowo przy pomocy polecenia *print (**p**)* lub w każdym kolejnym kroku po uprzednim zdefiniowaniu zmiennych do wyświetlenia poleceniem ***disp***.

(gdb) p <nazwa_zmiennej> - *wyświetla wartość zmiennej*
(gdb) p/<format> <nazwa_zmiennej>
- *wyświetla wartość zmiennej w żądanym formacie (help x – informacja o dostępnych formatach)*
(gdb) disp <nazwa_zmiennej> - *dodaje zmienną do wyświetlanej listy*
(gdb) undisplay <nr_wyswietlanej_zmiennej>
- *usuwa zmienną z wyświetlanej listy*

- Usunięcie wcześniej ustawionych pułapek wykonywane jest za pomocą polecenia *delete* (***d***).

(gdb) d <nr_kolejny_pułapki> - *usuwa wybraną pułapkę*
(gdb) d - *usuwa wszystkie ustawione pułapki*

- Aby przerwać działanie debugowanego programu należy wcisnąć kombinację *CTRL+C*
- Zakończenie pracy debugera GDB następuje po wydaniu polecenia *quit (**q**)*.

(gdb) q

g) Debugowanie programów napisanych w języku C, przeznaczonych na mikrokontroler Freescale MCF5282

- Po uruchomieniu debugera skrośnego debugowanie odbywa się w sposób analogiczny jak powyżej.
- Często przydatna jest możliwość wyświetlenia podczas wykonywania programu nazw rejestrów wewnętrznych mikrokontrolera wraz z ich zawartością, co jest realizowane za pomocą polecenia *info registers (**i r**)*.

(gdb) i r - *wyświetla zawartość wszystkich rejestrów dla aktualnej ramki stosu (tj. fragmentu stosu używa--nego przez funkcję)*
(gdb) i r <nazwa_rejestru> - *wyświetla zawartość tylko wybranego rejestru dla aktualnej ramki stosu*

- Drukowanie zawartości rejestru jest także możliwe przy użyciu polecenia *print (**p**)*.

(gdb) p $<nazwa_rejestru> - *wyświetla zawartość rejestru w zapisie dziesiętnym*
(gdb) p/<format> $<nazwa_rejestru>
- *wyświetla zawartość rejestru w żądanym zapisie (help x – informacja o dostępnych formatach)*

5. Korzystanie z oprogramowania dostępnego na laboratorium z przedmiotu SMCR

a) 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 debuger skrośny poleceniem *cfbdm* <*program.elf*>
   - Uruchomić program poleceniem *run* ***(r)***
   - Aby przerwać działanie programu należy wcisnąć kombinację *CTRL+C*
   - Aby zakończyć pracę z debugerem należy wpisać polecenie *quit* ***(q)***

b) Debugowanie programów napisanych w języku asemblera
   - Debugowanie odbywa się w podobny sposób jak w przypadku programów napisanych w języku C – patrz punkt 4f i 4g.