



Akademia Górniczo-
Hutnicza
w Krakowie
Katedra Elektroniki
WIET



Laboratorium TM2

Ćwiczenie 6

Port szeregowy UART0

Autor: Mariusz Sokołowski

wer. 21.11.2020

1. WSTĘP

1.1.CEL

Celem ćwiczenia jest:

- + zapoznanie studenta z techniką poprawnej inicjalizacji i obsługi portu szeregowego,
- + nabycie umiejętności łączenia funkcjonalnego portu szeregowego, w konfiguracji RS232C, z komputerem klasy IBM-PC,
- + poznanie możliwości wykorzystania terminala szeregowego (komputer) do:
 - sterowania peryferiami modułu FRDM-KL05Z,
 - obrazowania wyników pomiarowych, układów zaimplementowanych w oparciu o moduł FRDM-KL05Z,

1.2.WYMAGANIA

Sprzętowe:

- komputer klasy PC, spełniający wymagania sprzętowe aplikacji KEIL v5,
- zestaw FRDM-KL05Z

Programowe:

- system operacyjny Windows 7 lub wyższy,
- środowisko Keil / uVision 5 MDK-ARM

Doświadczenie:

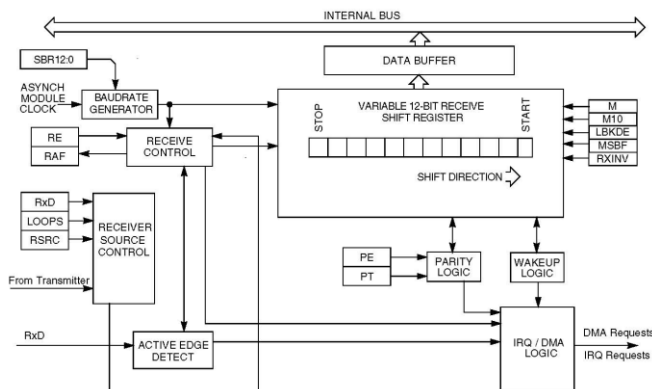
- podstawowa umiejętność obsługi komputera klasy IBM-PC,
- podstawowa znajomość systemów operacyjnych rodziny Windows,
- podstawowa znajomość zagadnień z Techniki Mikroprocesorowej 1
- podstawowa znajomość zagadnień z Techniki Cyfrowej
- poprzednie tematy laboratorium Techniki Mikroprocesorowej 2

Literatura:

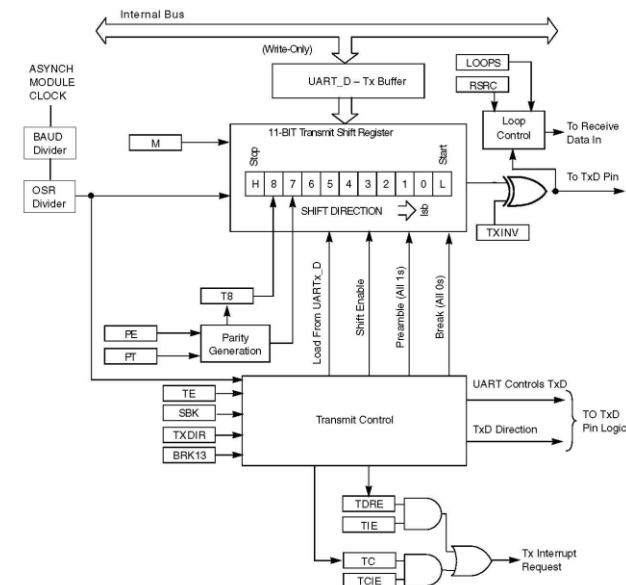
- KL05 Sub-Family Reference Manual, Freescale Semiconductor
- Kinetis L Peripheral Module Quick Reference, Freescale Semiconductor

2. MODUŁ PORTU SZEREGOWEGO UART0

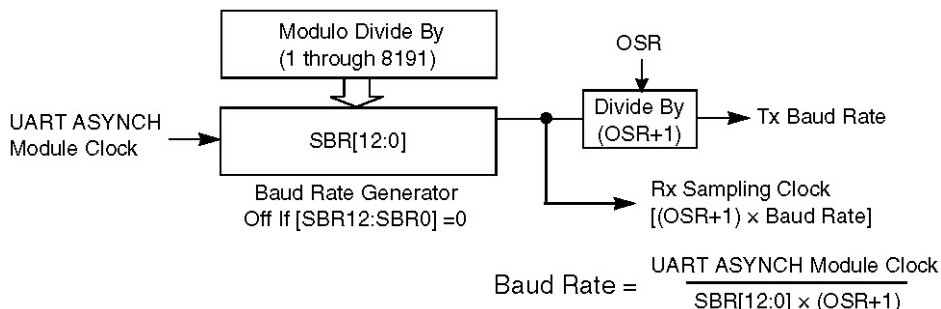
Układ MKL05Z32VLC4, będący centralnym mikrokontrolerem zestawu FRDM-KL05Z, posiada w swoich zasobach sprzętowych, jeden port szeregowy - UART0 (Rys. 1, 2 i 3).



Rys.1. Układ odbiornika UART0



Rys.2 Układ nadajnika UART0



Rys. 3. Generator taktujący nadajnik i odbiornik (Baud Generator)

Wybrane, najważniejsze cechy układu to:

- długość danej 8, 9 lub 10 bitów,
- możliwość nadawania i odbierania w tym samym czasie (Full-duplex),
- 13-bitowy dzielnik zegara taktującego, dający możliwość uzyskania różnych prędkości transmisji,
- sprzętowa generacja i weryfikacja bitu parzystości,
- wybór jednego lub dwóch bitów stopu,
- monitoring błędów:
 - Overrun Error - brak możliwości zapisania do bufora odbiornika nowej danej (skompletowanej w odbiorniku), spowodowany nieodczytaniem poprzedniej wartości z bufora,

- Noise Error - wykrycie niestaości stanu, dla danego, odczytywanego bitu,
- Framing Error - bład ramki, spowodowany wykryciem stanu „0” dla bitu stopu, który powinien mieć wartość „1”,
- Parity Error - wartość otrzymanego bitu parzystości nie zgadza się z zawartością danej.
- „oversampling” (próbkiowanie nadmiarowe) - zwielokrotnianie częstotliwości próbkowania w odbiorniku, w stosunku do generatora taktującego (Baud Generator) - Rys. 3.

Prędkość transmisji BR jest kreowana w oparciu o częstotliwość wybranego zegara taktującego (SIM_SOPT2[UART0SRC]) oraz zawartość rejestrów UART0_BDH[SBR], UART0_BDL[SBR] i UART0_C4 [OSR]. Wielkości te są powiązane następującym wzorem:

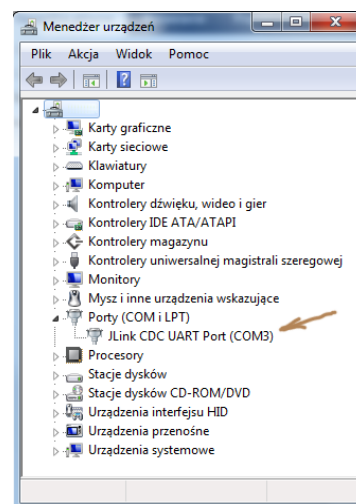
$$BR = \frac{UART0SRC}{(OSR + 1) * SBR}$$

Aby przygotować układ UART0 do pracy, należy wykonać następujące czynności:

- ❖ Sprawdzić, jaką wartość ma stała CLOCK_SETUP, w zbiorze *system_MKL05Z4.c*. Informacja ta będzie miała wpływ na ustawienia rejestrów, odpowiedzialnych za szybkość transmisji. W zależności od stałej CLOCK_SETUP, parametry podstawowych sygnałów zegarowych mają następujące wartości:
 - CLOCK_SETUP=0 (wartość domyślna):
 - zegar referencyjny dla modułu MCG - 32768Hz,
 - Core clock - 41943040Hz,
 - BusClock - 20971520Hz.
 - CLOCK_SETUP=1:
 - zegar referencyjny dla modułu MCG - 32768Hz,
 - Core clock - 47972352Hz,
 - BusClock - 23986176Hz.
 - CLOCK_SETUP=2:
 - zegar referencyjny dla modułu MCG – 4MHz,
 - Core clock – 4MHz,
 - BusClock – 2MHz.
- ❖ dołączyć sygnał taktujący do modułu UART0, w rejestrze SIM_SCGC4 [UART0=1],
- ❖ dołączyć sygnał taktujący do odpowiedniego portu, którego końcówki realizują funkcje TX (PTB1) i RX (PTB2), w rejestrze SIM_SCGC5[PORTB=1],
- ❖ ustawić odpowiednią funkcję dla wykorzystywanych końcówek portu, w rejestrze PORTB_PCRx[MUX=2]. PTB1 - nadajnik TX, PTB2 - odbiornik RX,
- ❖ ustawić źródło zegara taktującego moduł UART0, w rejestrze SIM_SOPT2[UART0SRC=1], MCGFLLCLK=Core clock (dla CLOCK_SETUP= 0 lub 1),

- ❖ zablokować nadajnik i odbiornik, w rejestrze UART0_C2[RE=0, TE=0],
- ❖ ustawić wartość dzielnika próbkowania nadmiarowego, w rejestrze UART0_C4 [OSR]:
 - BR=28800: CLOCK_SETUP=0, OSR=15,
 - BR=230400: CLOCK_SETUP=1, OSR=15.
- ❖ ustawić możliwość próbkowania w odbiorniku, danych przychodzących, na obydwu zboczach zegara, w rejestrze UART0_C5[BOTHEDGE=1], w przypadku, gdy wartość OSR jest wybrana z przedziału od 3 do 6 (patrz powyżej). Dla innych ustawień OSR, opcja ta leży w gestii programisty,
- ❖ ustawić 13-bitową wartość dzielnika, będącego źródłem zegara dla odbiornika i nadajnika. Najpierw starsze 5 bitów ustawić w rejestrze UART0_BDH[SBR], a następnie młodsze 8 bitów w rejestrze UART0_BDL[SBR] - wziąć pod uwagę nastawioną wartość OSR oraz wzór na BR:
 - BR=28800: CLOCK_SETUP=0, BDH[SBR=0], BDL[SBR=91],
 - BR=230400: CLOCK_SETUP=1, BDH[SBR=0], BDL[SBR=13].
- ❖ ustawić jeden bit stopu, w rejestrze UART0_BDH[SBNS=0],
- ❖ ustawić długość danej na 8 bitów oraz brak sprzętowej obsługi sprawdzania parzystości, w rejestrze UART0_C1[M=0, PE=0],
- ❖ w zależności od potrzeb, włączyć przerwania od nadajnika i/lub odbiornika, w rejestrze UART0_C2[TIE=1 i/lub RIE=1]. Nadajnik zgłasza przerwanie, gdy bufor nadajnika jest pusty, a odbiornik, gdy bufor odbiornika jest pełny. Jeżeli chodzi o nadajnik, to jest jeszcze jedna możliwość zgłaszania przerwania: gdy nadajnik skończył transmisję (UART0_C2[TCIE=1]),
- ❖ włączyć nadajnik i odbiornik, w rejestrze UART0_C2[TE=1, RE=1].

W tym momencie układ UART0 jest gotowy do pracy. Nadawanie polega na wpisywaniu danej 8-bitowej do rejestru UART0_D, a odbiór, to po prostu odczyt tego rejestru. Jeśli nie używamy przerw do komunikacji z nadajnikiem i odbiornikiem, stan nadajnika sprawdzamy poprzez odczyt wartości bitu TDRE w rejestrze UART0_S1, a stan odbiornika poprzez odczyt wartości bitu RDRF w tymże rejestrze. TDRE=1 – rejestr nadajnika pusty, RDRF=1 – odbiornik pełny. W przypadku wybrania opcji zgłaszania końca transmisji (TCIE), sprawdzany jest bit UART0_S1[TC], TC=1 – transmisja zakończona. Jeżeli nadajnik nie jest gotowy na następną daną, nie należy jej tam zapisywać.



Rys.4.

Od strony komputera należy sprawdzić, uruchamiając menedżera urządzeń, który port COM jest przypisany do naszego modułu (kabel USB dołączony do złącza OpenSDA) - Rys. 4. Pisząc oprogramowanie lub łącząc się poprzez terminal, właśnie tego numeru portu COM należy używać. Od strony modułu, komunikacja jest zapewniona poprzez dołączone

końcówki PTB1 i PTB2 do mikrokontrolera PK20DX128VFM5, realizującego interfejs Open-SDA.

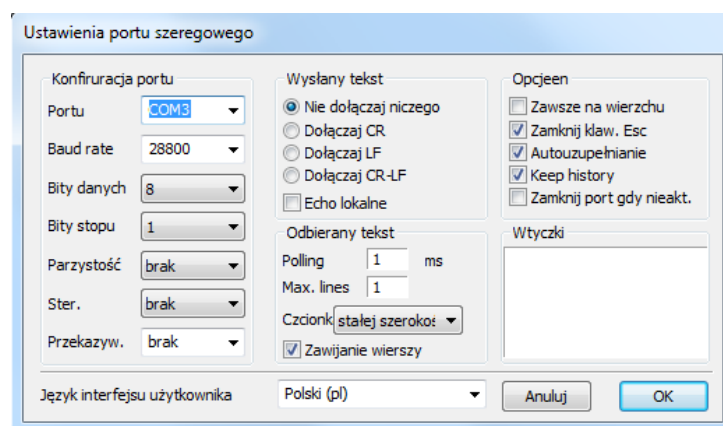
Ponieważ istnieje jeden wektor obsługujący port UART0, w podprogramie obsługi należy najpierw sprawdzić, które z urządzeń, nadajnik czy odbiornik, jest źródłem przerwania. Jeśli są również włączone przerwania od błędów, to krąg „sprawców” się powiększa. Po ustaleniu źródła przerwania, odpowiednia flaga, w rejestrze UART_S1 powinna zostać skasowana. Flaga RDRF (od odbiornika) kasowana jest automatycznie, w chwili odczytu bufora danych odbiornika (UART0_D). Flagi TDRE i TC (od nadajnika) są kasowane automatycznie, w chwili zapisu bufora nadajnika (UART0_D). Pozostałe flagi należy wyzerować programowo, wpisując w odpowiednią pozycję rejestru UART0_S1 wartość „1”.

3. ĆWICZENIE NR 1 – NAWIAZYWANIE POŁĄCZENIA Z KOMPUTEREM

Rozpakować zbiór *Lab_6.zip*. Zbiór zawiera również katalog z prostym programem terminalowym. Jego rozszerzenie należy zmienić na *.exe*.

Ponieważ w ćwiczeniach będą używane funkcje operujące na wartościach zmiennoprzecinkowych, zwiększono pojemność stosu, ustawiając go na wartość: **Stack_Size EQU 0x00000300**, w zbiorze *startup_MKL05Z4.s*.

Uruchomić projekt *Cw_1.uvprojx*, a na komputerze program terminala *Termite.exe*. Ustawienia powyższego programu są przedstawione na rysunku Rys. 5.



Rys. 5. Ustawienia programu *Termite*

Po nawiązaniu połączenia, program wyświetla, co ok. 0.5s, kolejne liczby od 0 do 255, wysyłane, w pętli, przez nadajnik UART0. Przeanalizować etapy programowania portu UART0 (zbiór *uart0.c*)

Uwaga:

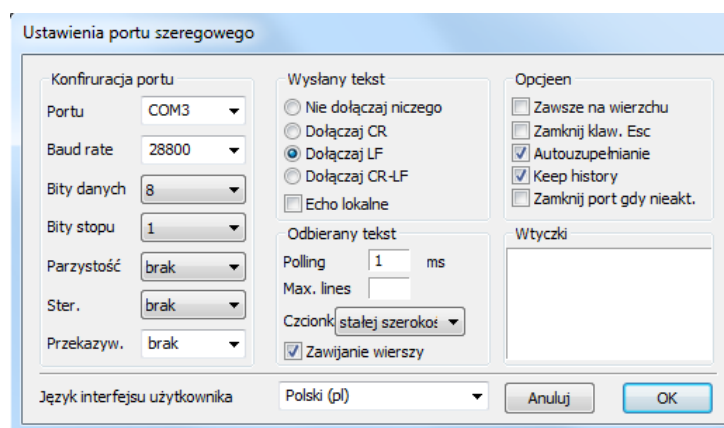
Program *Termite.exe* może po pewnym czasie zgłosić błąd i przestać działać. Należy go ponownie uruchomić. Jednak jego podstawowe zalety, to możliwość ustawienia jednej linii wyświetlania oraz reagowanie na znak 0xD (powrót karetki do początku linii).

Zadanie domowe:

Podłączyć czujnik światła wg punktu 4, z instrukcji do laboratorium 5, „Przetwornik A/C i C/A”. Odpowiednio uzupełnić i zmodyfikować projekt *Cw_1* tak, aby teraz zamiast kolejnych liczb, wyświetlał się, co ok. 0.5 sekundy, wynik pomiaru napięcia wyjściowego czujnika światła, w postaci: $U=0.0128V$. Można zastosować wersję bez uśredniania. Wynik wyświetlić równoległe również na wyświetlaczu LCD, dla weryfikacji.

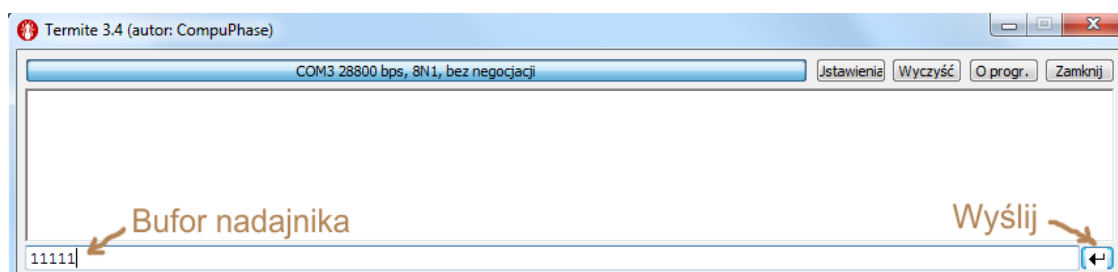
4. ĆWICZENIE NR 2 – TESTOWANIE ODBIORNIKA, W TRYBIE Z PRZERWANIAMI

Uruchomić projekt *Cw_2.uvprojx* i program *Termite.exe*. Ustawienia powyższego programu są przedstawione na rysunku Rys. 6.



Rys. 6. Ustawienia programu *Termite*

W polu „Bufor nadajnika”, programu *Termite*, wpisać dowolny ciąg znaków (Rys. 7). Po wciśnięciu klawisza „Wyślij”, znaki z bufora nadajnika zostaną odebrane przez odbiornik modułu i zwrótnie wysłane z powrotem do odbiornika programu *Termite*. Przeanalizować działanie projektu.



Rys. 7. Wysyłanie za pomocą programu *Termite*

5. ĆWICZENIE NR 3 – STEROWANIE URZĄDZENIAMI ZEWNĘTRZNYMI ZA POMOCĄ TERMINALA

Uruchomić projekt *Cw_3.uvprojx* i program *Termite.exe*, którego ustawienia są takie same jak w pkt. 4.

Program, w zależności od przesłanej z terminala komendy, zapala lub gasi czerwoną diodę LED. Komenda „LRON” zapala, a „LROFF” gasi. Podanie złej komendy powoduje wyświetlenie, w oknie odbiornika terminala, komunikatu „Zła komenda”. Podanie zbyt długiego ciągu znaków (większego od 15) powoduje wyświetlenie, w oknie odbiornika terminala, komunikatu „Zbyt długi ciąg”.

Zadanie domowe:

Zrealizować sterowanie pozostałymi diodami LED.

Zadanie dodatkowe:

Wykorzystując sterowanie jasnością czerwonej diody LED, za pomocą PWM (patrz instrukcja do laboratorium 4, „Moduły liczników TPMx - ich obsługa i sposób wykorzystania”), zrealizować projekt, który pozwoli zadawać jasność świecenia z terminala.