



Politechnika Opolska

LABORATORIUM

Technika Mikroprocesorowa

KIERUNEK STUDIÓW:	AiR Ns		ROK STUDIÓW:	III
SEMESTR:	VI	ROK AKADEMICKI:	2019/2020	

Temat ćwiczenia:

Komunikacja szeregową UART

Projekt wykonali:

Nazwisko i imię:		Nazwisko i imię:	
1.	Sebastian Juraszek	2.	

Ocena:	Data:	Uwagi:

1. Wstęp

Jeżeli kontroler pracujący w urządzeniu musi komunikować się z innym kontrolerem, to najwygodniej zrobić to wykorzystując jeden z przyjętych standardów komunikacyjnych. Interfejs szeregowy UART umożliwia dwukierunkową wymianę danych przesyłając dane kolejno, bit po bicie.

TX – linia danych wysyłanych

RX – Linia danych odbieranych

GND - linia masy, ustalająca poziom odniesienia dla połączonych urządzeń

Należy pamiętać aby sygnał wysyłany z jednego kontrolera linią danych TX, był połączony z drugim kontrolerem z linią danych RX.

Zasada działania interfejsu

Transmisję informacji złączem szeregowym inicjuje tzw. bit startowy. W każdym przypadku jest to logiczne zero. W dalszej kolejności przesłana zostaje informacja w postaci 7, 8 lub 9 kolejnych bitów (w zależności od ustalonej konfiguracji urządzenia). Za zakończenie transmisji odpowiedzialny jest tzw. bit stopu, który zawsze jest równy logicznej jedynce. Całość tworzy tzw. ramkę UART, która zawiera w sobie kompletną informację.

2. Opis

Celem ćwiczenia było wykonanie komunikacji szeregowej UART poprzez linię danych TX, na linię danych TX tego samego kontrolera. Program ma za zadanie co sekundę a TX wystawić dowolną liczbę natomiast na RX ma być o 10 większa. Po przekroczeniu 30 licznik ma się wyzerować i liczyć od zera.

3. Skrypt programu:

```
4. //Zadanie 4
5.
6.
7.
8. #include <msp430.h>
9.
10. void print(char *nazwa)
11. {
12.     unsigned int i = 0;
13.     while(nazwa[i] != '\0')
14.     {
15.         while (!(IFG2&UCA0TXIFG));
```

```

15.         UCA0TXBUF = nazwa[i];
16.         i++;
17.     }
18. }
19.
20. void printNumber(unsigned int pmc)
21. {
22.     char buf[6];
23.     char *str = &buf[5];
24.
25.     *str = '\0';
26.
27.     do
28.     {
29.         unsigned long m = pmc;
30.         pmc /= 10;
31.         char c = (m - 10 * pmc) + '0';
32.         *--str = c;
33.     } while(pmc);
34.
35.     print(str);
36. }
37.
38. void main(void)
39. {
40.     WDTCTL = WDTPW + WDTHOLD;           // Stop Watchdog
41.     if (CALBC1_1MHZ==0xFF)             // Check if calibration constant
erased
42.     {
43.         while(1);                       // do not load program
44.     }
45.     DCOCTL = 0;                         // Select lowest DCO settings
46.     BCSCTL1 = CALBC1_1MHZ;              // Set DCO to 1 MHz
47.     DCOCTL = CALDCO_1MHZ;
48.
49.     P1SEL = BIT1 + BIT2 ;               // Select UART RX/TX function on
P1.1,P1.2
50.     P1SEL2 = BIT1 + BIT2;
51.
52.     UCA0CTL1 |= UCSSEL_2;               // UART Clock -> SMCLK
53.     UCA0BR0 = 104;                     // Baud Rate Setting for 1MHz
9600
54.     UCA0BR1 = 0;                       // Baud Rate Setting for 1MHz
9600
55.     UCA0MCTL = UCBRS_1;                 // Modulation Setting for 1MHz
9600
56.     UCA0CTL1 &= ~UCSWRST;               // Initialize UART Module
57.     IE2 |= UCA0RXIE;                   // Enable RX interrupt
58.
59.     __bis_SR_register(LPM0_bits + GIE); // Enter LPM0, Enable Interrupt
60. }
61.
62. #pragma vector=USCIAB0RX_VECTOR         // UART RX Interrupt
Vector
63. __interrupt void USCI0RX_ISR(void)
64. {
65.     while (!(IFG2&UCA0TXIFG));         // Check if TX is
ongoing

```

```
66.          UCA0TXBUF = UCA0RXBUF;          // TX -> Received
    Char + 1
67.
68.    printNumber(UCA0TXBUF);
69.    print("\tJEST OK\r\n");
70.
71. }
```

4. Wnioski

Komunikacja szeregową UART jest dużym ułatwieniem, ponieważ umożliwia nam wymianę danych pomiędzy 2 mikrokontrolerami lub komputerem. Komunikację można wykorzystać przy budowie większych projektów, gdzie trzeba wymieniać się informacjami. Ćwiczenie 4 nie należało do najłatwiejszych, dlatego wszelkie niejasności konsultowałem z kolegami z grupy.