



Politechnika Opolska

LABORATORIUM

PRZEDMIOT: **Technika mikroprocesorowa**

KIERUNEK STUDIÓW: **Automatyka i Robotyka** ROK STUDIÓW: **III**

SPECJALNOŚĆ: **-**

SEMESTR: **VI** ROK AKADEMICKI: **2019/2020**

Nr ćwiczenia: **4**

Temat ćwiczenia:

Komunikacja UART - MSP430.

Ćwiczenie wykonali:

Nazwisko:

Imię:

Nazwisko:

Imię:

1.	Dziembowski	Mateusz	2.	--	--
3.	--	--	4.		

<u>Uwagi:</u>	<u>Data:</u>	<u>Ocena za sprawozdanie:</u>

1. Założenia

Wykorzystując zestaw MSP-EXP430G2 produkcji Texas Instruments, napisać program umożliwiający nawiązanie cyfrowej komunikacji - poprzez szeregowy interfejs UART. Aby poprawnie przejść przez proces należy zmienić ustawienie zworek na pinach TXD oraz RXD wg instrukcji urządzenia. Zostanie wybudowany kanał komunikacji poprzez złącze USB z odczytem komunikatu w programie PuTTY. W składni programu znajdzie się treść oraz inkrementowana liczba, które n-razy zostaną przesłane na ekran komputera.

2. Opis działania programu

```

33 void main(void)
34 {
35     WDTCTL = WDTPW + WDTHOLD;
36     if (CALBC1_1MHZ == 0xFF)          // sprawdzenie kalibracji
37     {
38         while (1);                    //
39     }
40     DCOCTL = 0;                        // nastawa DCO Clock Frequency Control
41     BCSC1L1 = CALBC1_1MHZ;            // nastawa DCO na 1MHz Basic Clock System Control 1
42     DCOCTL = CALDCO_1MHZ;            // DCO Clock Frequency Control
43
44     P1SEL = BIT1 + BIT2;              // ustawienie Rx/Tx na P1.1,P1.2
45     P1SEL2 = BIT1 + BIT2;
46
47     UCA0CTL1 |= UCSSEL_2;             // USCIA0 Control Register 1
48     UCA0BR0 = 104;                   // USCIA0 Baud Rate 0
49     UCA0BR1 = 0;                     // USCIA0 Baud Rate 1
50     UCA0MCTL = UCBRS_1;              // USCIA0 Modulation Control
51     UCA0CTL1 &= ~UCSWRST;            // USCIA0 Control Register 1
52
53     unsigned int c = 0;
54     do
55     {
56         c++;
57         wyswietl_liczbe(c);           // wypisz liczbe
58         wyswietl_tekst("\t oczy mnie bola\r\n"); // wypisz dany tekst
59     }
60     while (c<=2);
61
62
63 }

```

Główna część skryptu zawarta jest w liniach 33-63. Dla poprawności komunikacji niezbędne jest wykonanie nastaw połączenia jak BaudRate czy Modulation. Niezbędne kalibracje zewarte są w liniach 36-51. Na ekranie komputera zostanie 3 razy wyświetlona liczba c oraz tekst oczy mnie bola z linii 57. Wyświetlanie liczby i ciągu znaków wykonane zostanie funkcjami **wyswietl_liczbe** oraz **wyswietl_tekst**. Opisany fragment zawarty jest pomiędzy liniami 33-63.

```

1 #include <msp430.h>
2
3 /* ----- wypisywanie liczby ----- */
4 void wyswietl_liczbe(unsigned int liczba)
5 {
6     char buf[6];
7     char* str = &buf[5];
8
9     *str = '\0';
10
11     do
12     {
13         unsigned long m = liczba;
14         liczba /= 10;
15         char c = (m - 10 * liczba) + '0';
16         *--str = c;
17     } while (liczba);
18
19     wyswietl_tekst(str);
20 }

```

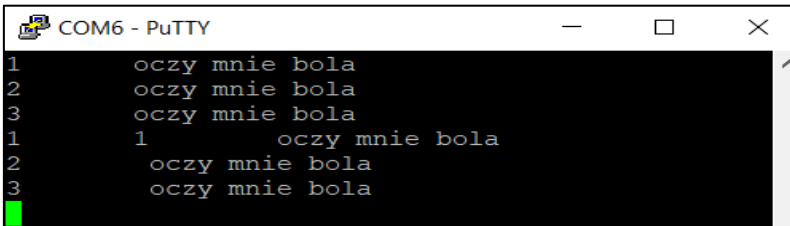
Zanim zmienna typu int zostanie wyświetlona na ekranie - zostanie przekonwertowana na znak i za pomocą funkcji **wyswietl_tekst** przedstawiona na zewnątrz. Funkcja konwersji i wyświetlania liczby całkowitej przedstawiona jest w liniach 3-20.

```

21 /* ----- wypisywanie tekstu ----- */
22 void wyswietl_tekst(char* text)
23 {
24     unsigned int i = 0;
25     while (text[i] != '\0')
26     {
27         while (!(IFG2 & UCA0TXIFG)); // sprawdzenie czy TX nie jest zajęte
28         UCA0TXBUF = text[i];         //
29         i++;
30     }
31 }

```

Funkcja odpowiedzialna za wyświetlanie ciągu znaków zawarta jest pomiędzy linią 22 a linią 31 skryptu.



```

1      oczy mnie bola
2      oczy mnie bola
3      oczy mnie bola
1      1      oczy mnie bola
2      2      oczy mnie bola
3      3      oczy mnie bola

```

Efekt działania opisanego skryptu.

3. Pełen kod programu

```

/* PROGRAM KOMUNIKACJI UART 06062020*/
/* AiR_ns 2019/2020 */
/* Mateusz Dziembowski */

#include <msp430.h>

/* ----- wypisywanie liczby-----*/
void wyswietl_liczbe(unsigned int liczba)
{
    char buf[6];
    char* str = &buf[5];

    *str = '\0';

    do
    {
        unsigned long m = liczba;
        liczba /= 10;
        char c = (m - 10 * liczba) + '0';
        *--str = c;
    } while (liczba);

    wyswietl_tekst(str);
}

/* ----- wypisywanie tekstu -----*/
void wyswietl_tekst(char* text)
{
    unsigned int i = 0;
    while (text[i] != '\0')
    {
        while (!(IFG2 & UCA0TXIFG)); // sprawdzenie czy TX nie jest zajęte
        UCA0TXBUF = text[i]; //
        i++;
    }
}

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD;
    if (CALBC1_1MHZ == 0xFF) // sprawdzenie kalibracji
    {
        while (1); //
    }
    DCOCTL = 0; // nastawa DCO Clock Frequency Control
    BCSCTL1 = CALBC1_1MHZ; // nastawa DCO na 1MHz Basic Clock System Control 1
    DCOCTL = CALDCO_1MHZ; // DCO Clock Frequency Control

    P1SEL = BIT1 + BIT2; // ustawienie Rx/Tx na P1.1,P1.2
    P1SEL2 = BIT1 + BIT2;

    UCA0CTL1 |= UCSSEL_2; // USCI A0 Control Register 1
    UCA0BR0 = 104; // USCI A0 Baud Rate 0
    UCA0BR1 = 0; // USCI A0 Baud Rate 1
    UCA0MCTL = UCBSR_1; // USCI A0 Modulation Control
    UCA0CTL1 &= ~UCSWRST; // USCI A0 Control Register 1

    unsigned int c = 0;
    do
    {
        c++;
        wyswietl_liczbe(c); //wypisz liczbe
        wyswietl_tekst("\t czy mnie bola\r\n"); //wypisz dany tekst
    } while (c<=2);
}

```