



Politechnika Opolska

LABORATORIUM

PRZEDMIOT:	Technika mikroprocesorowa
-------------------	---------------------------

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

<i>Temat ćwiczenia:</i>	
	Zadanie 4

<i>Projekt wykonali:</i>			
<i>Nazwisko i imię:</i>		<i>Nazwisko i imię:</i>	
1.	Marek Kaczmarczyk	2.	Dariusz Woźnica
3.	Michał Wacławczyk	4.	

<i>Ocena:</i>	<i>Data:</i>	<i>Uwagi:</i>

Opis zadania.

Program do obsługi transmisji UART.

Wystawia co sekundę inkrementowaną liczbę.

Gdy na porcie rx liczba będzie równa 30 .

Liczba jest zerowana, cykl się powtarza.

```
#include <msp430.h>
#include <stdlib.h>
void print(char *text)
{
    unsigned int i = 0;
    while(text[i] != '\0')
    {
        while (!(IFG2&UCA0TXIFG)); // sprawdzenie czy można wystawic znak
        UCA0TXBUF = text[i];      // wystawienie znaku na tx
        i++;
    }
}

void printNumber(unsigned int num) // zamiana int na string
{
    char buf[6];
    char *str = &buf[5];

    *str = '\0';

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

    print(str); // wywołanie funkcji wystawienia na tx

__interrupt void USCI0RX_ISR(void)
{
    char buf[6];
    char *recieved_str = &buf[5];
    recieved_str = UCA0RXBUF;
    if (recieved_string == '30') // '30' otrzymano?
    {
        count = 0; // zeruj wysyłaną liczbę
    }
    recieved_number = atoi(str)+10; // dodanie 10 do przekonwertowanego na int stringa otrzymanego z
    pinu rx
```

```

}

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD;    // Stop Watchdog
    if (CALBC1_1MHZ==0xFF)        // Check if calibration constant erased
    {
        while(1);                // do not load program
    }
    DCOCTL = 0;                  // Select lowest DCO settings
    BCSCTL1 = CALBC1_1MHZ;        // Set DCO to 1 MHz
    DCOCTL = CALDCO_1MHZ;

    P1SEL = BIT1 + BIT2 ;        // Select UART RX/TX function on P1.1,P1.2
    P1SEL2 = BIT1 + BIT2;

    UCA0CTL1 |= UCSSEL_2;         // UART Clock -> SMCLK
    UCA0BR0 = 104;                // Baud Rate Setting for 1MHz 9600
    UCA0BR1 = 0;                 // Baud Rate Setting for 1MHz 9600
    UCA0MCTL = UCBRS_1;           // Modulation Setting for 1MHz 9600
    UCA0CTL1 &= ~UCSWRST;         // Initialize UART Module

    UCA0IE |= UCA0RXIE;          // Enable USCI_A0 RX interrupt

    unsigned int count = 0;
    __bis_SR_register(LPM04 + GIE);
    while(1)
    {

        printNumber(count);
        count++;
        __delay_cycles(10000);

    }

}

```