



Politechnika Opolska

LABORATORIUM

Technika Mikroprocesorowa

KIERUNEK STUDIÓW:	Automatyka i Robotyka	ROK STUDIÓW:	III
SEMESTR:	VI	ROK AKADEMICKI:	2019/2020

Temat ćwiczenia:

UART

Projekt wykonali:

Nazwisko i imię:		Nazwisko i imię:	
1.	Leszek Cieśla	2.	Ryszard Hałapacz
3.	-----	4.	-----

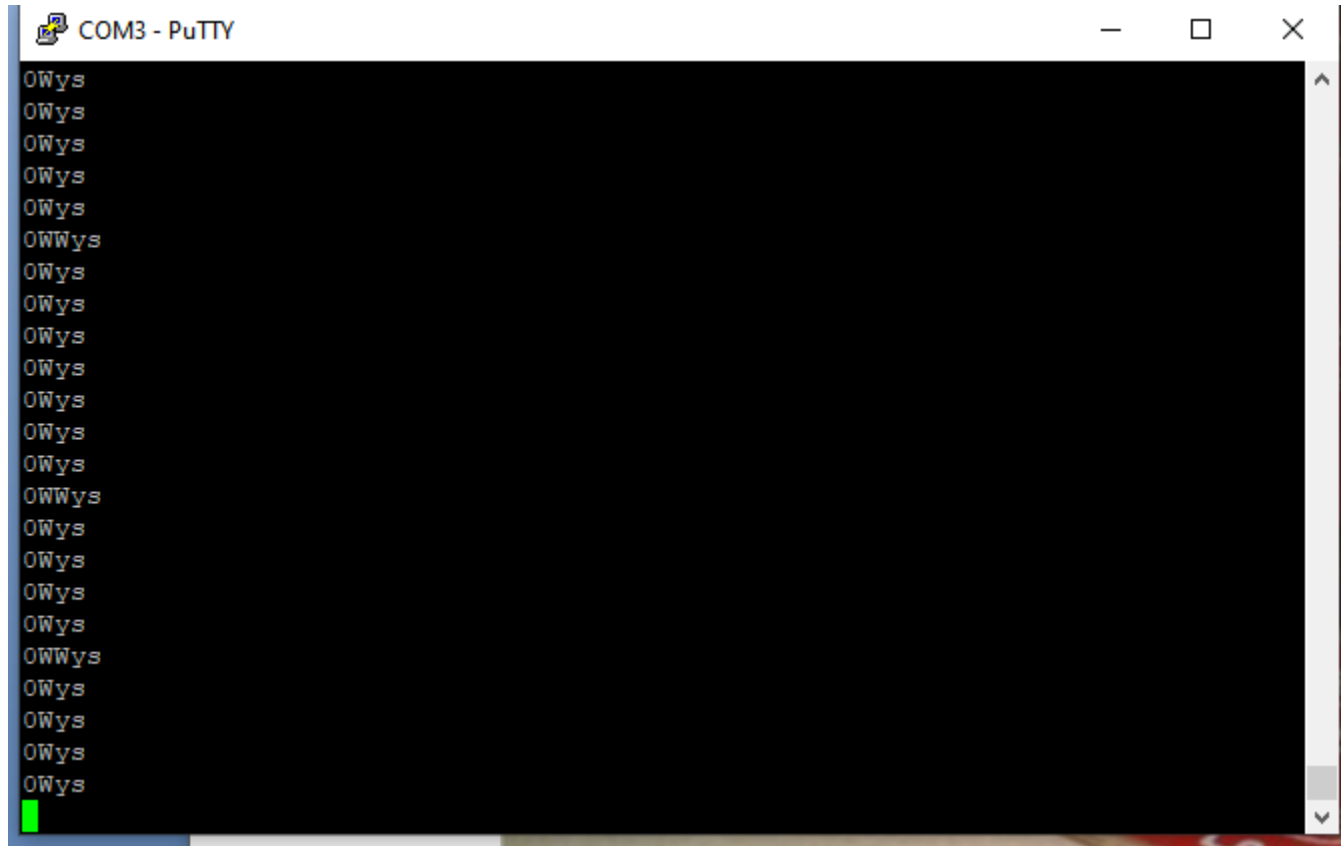
Ocena:	Data:	Uwagi:

1. Wstęp teoretyczny.

W mikrokontrolerze MSP430 komunikacja szeregową jest obsługiwana przez układ peryferyjny o nazwie USCI. Urządzenie to zostało skonfigurowane by móc obsługiwać wiele formatów komunikacji szeregowej. Zastosujemy komunikację szeregową zwaną jako UART. Jest to tryb wykorzystujący dwa piny do przesyłania (UCA0TXD) i odbierania (UCA0RXD). Asynchroniczna komunikacja jest zaprzęgnięta do komunikacji z urządzeniami zewnętrznymi takimi jak komputer. W tym laboratorium zastosujemy mikrokontroler do tego celu. Program ma 2 rzeczy do zrobienia: transmisję i wysyłanie danych.

Program putty posłużył nam za terminal do komunikacji z mikrokontrolerem.

Program ma obsługę zdarzeń transmisji i odbierania danych. Jeżeli odbiera dane, wysyła chara – O, jeżeli wykonuje transmisję na wyjście idzie tablica charów message o treści – Wys. Poniższy screen dowodzi obsługi obydwu zdarzeń. Ponadto mikrokontroler reaguje na obsługę zdarzeń zmianą stanu diod LED.



2. Skrypt napisany w języku C

```
#include "msp430g2553.h"
char message[] = { "Wys \r\n" };
char message2[] = { "Odb \r\n" };
char message3[] = { "123 \r\n" };
int flag = 0;
```

```

void print(char *text)
{
    unsigned int i = 0;
    while(text[i] != '\0')           // petla do konca stringa
    {
        while (!(IFG2&UCA0TXIFG));   // Check if TX is ongoing
        UCA0TXBUF = text[i];         // wysylanie znaku
        i++;                          // inkrementuj
    }
}

void main(void)
{
    WDTCTL = WDTPW + WDTHOLD; // Stop the Watch dog

    //----- Configure the Clocks -----//

    if (CALBC1_1MHZ==0xFF) // If calibration constant erased
    {
        while(1);           // nie wczytywac
    }

    DCOCTL = 0;              // Select lowest DCOx and MODx settings
    BCSCTL1 = CALBC1_1MHZ;   // Set range
    DCOCTL = CALDCO_1MHZ;   // Set DCO step + modulation

    //----- Configuring the LED's -----//

    P1DIR |= BIT0 + BIT6;    // P1.1 UCA0RXD wejście
    P1OUT &= ~BIT0 + BIT6;   // Pwyjście P1.2 UCA0TXD

    //----- Ustawienie funkcji UART dla P1.1 & P1.2 -----//

    P1SEL |= BIT1 + BIT2;    // P1.1 UCA0RXD input
    P1SEL2 |= BIT1 + BIT2;   // P1.2 UCA0TXD output

    //----- Konfigurowanie UART (USCI_A0) -----//

    UCA0CTL1 |= UCSSEL_2 + UCSWRST; // Zegar USCI = SMCLK, USCI_A0 wyłączone
    UCA0BR0 = 104;                // 1104 Z tabeli arkusza danych
    UCA0BR1 = 0;                  // selects baudrate = 9600, clk = SMCLK
    UCA0MCTL = UCBRS_1;           // Wartość modulacji = 1 z arkusza danych
    UCA0STAT |= UCLISTEN;         // włączony tryb petli zwrotnej
    UCA0CTL1 &= ~UCSWRST;         // Wyczyść UCSWRST, aby włączyć USCI_A0

    //----- Enabling the interrupts -----//

    IE2 |= UCA0TXIE;             // Włącz przerwanie transmisji
    IE2 |= UCA0RXIE;             // Włącz przerwanie odbioru
    _BIS_SR(GIE);                // Włącz globalne przerwanie

    UCA0TXBUF='0';
    // print(message);            // Transmit a byte

    _BIS_SR(LPM0_bits + GIE);    // Going to LPM0
}

//-----//
// // Przesyłaj i odbieraj przerwania // //

```

```
//-----//

#pragma vector = USCIAB0TX_VECTOR
__interrupt void TransmitInterrupt(void)
{ // ;
    //print(message);
    UCA0TXBUF='0';
    delay_cycles(50000);
    P1OUT ^= BIT0; //świeci P1.0 Dioda na Tx
}

#pragma vector = USCIAB0RX_VECTOR
__interrupt void ReceiveInterrupt(void)
{
    print(message2);

    P1OUT ^= BIT6;    // świeci dioda P1.6 na RX
    IFG2 &= ~UCA0RXIFG; // wyczyśc flage
}

```

3. Wnioski.

Udało nam się skomunikować płytke z komputerem. Zostało to osiągnięte za pomocą asynchronicznej komunikacji szeregowej Tryb ten wykorzystuje dwa piny do przesyłania i odbioru danych Trzeba było skonfigurować MSP430, poprzez przełączenie zworek. Aby użyć portu szeregowego Launchpad, należało najpierw skonfigurować zworki na płycie. W pobliżu sekcji Emulator znajduje się 5 zworek łączących sekcję Emulator z układem MSP430G2553. Należy ustawić piny RXD i TXD w sprzętowym trybie UART (HW UART), W tym ćwiczeniu potrzebne też było dokończenie konfiguracji po stronie PC poprzez instalację dodatkowego oprogramowania i konfigurację łącza poprzez wprowadzanie numeru Com odpowiadającego szeregowemu portowi a także należało ustawić szybkość transmisji..

