|  |  |
| --- | --- |
|  | **Politechnika Opolska**  **L A B O R A T O R I U M** |

|  |  |
| --- | --- |
| Przedmiot: |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Kierunek studiów: | **AiR** | | Rok studiów: | | **III** |
| Specjalność: | **-** | |  | | |
| Semestr: | **VI** | Rok akademicki: | | ***2019/2020*** | |

|  |  |  |
| --- | --- | --- |
| *Nr ćwiczenia:* | **4** |  |

|  |
| --- |
| *Temat ćwiczenia:* |
|  |
|  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Ćwiczenie wykonali:* | | | | | |
| *Nazwisko:* | | *Imię:* | *Nazwisko:* | | *Imię:* |
| 1. | Baldy | Dominik | 2. | Niewiora | Mateusz |
| 3. | Lazik | Michael | 4. |  |  |

|  |  |  |
| --- | --- | --- |
| *Uwagi:* | *Data:* | *Ocena za sprawozdanie:* |
|  |  |  |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Termin zajęć: | | | | | |
| Data: | 5.04.2020 | Dzień tygodnia: | Sobota | Godzina: | 10:00 |

|  |  |  |  |
| --- | --- | --- | --- |
| Termin oddania sprawozdania: | 10.04.2020 | Sprawozdanie oddano: |  |

1. Założenie projektu

Program powinien korzystać z modułu połączenia między 2 płytkami MPS430 i komunikować się między nimi. Płytka odbierająca po otrzymaniu liczby z płytki nadającej inkrementuje tą liczbe o 10 i dodaje ją do licznika, po przekroczeniu 30 licznik resetuje się.

1. Kod:

#include <msp430.h>

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

IE2 |= UCA0RXIE; // Enable RX interrupt

\_\_bis\_SR\_register(LPM0\_bits + GIE); // Enter LPM0, Enable Interrupt

}

#pragma vector=USCIAB0RX\_VECTOR // UART RX Interrupt Vector

\_\_interrupt void USCI0RX\_ISR(void)

{

while (!(IFG2&UCA0TXIFG)); // Check if TX is ongoing

UCA0TXBUF =UCA0TXBUF + UCA0RXBUF + 10; // dodaj 10 do otrzymanej

if(UCA0TXBUF => 30)'

UCA0TXBUF=0;

}