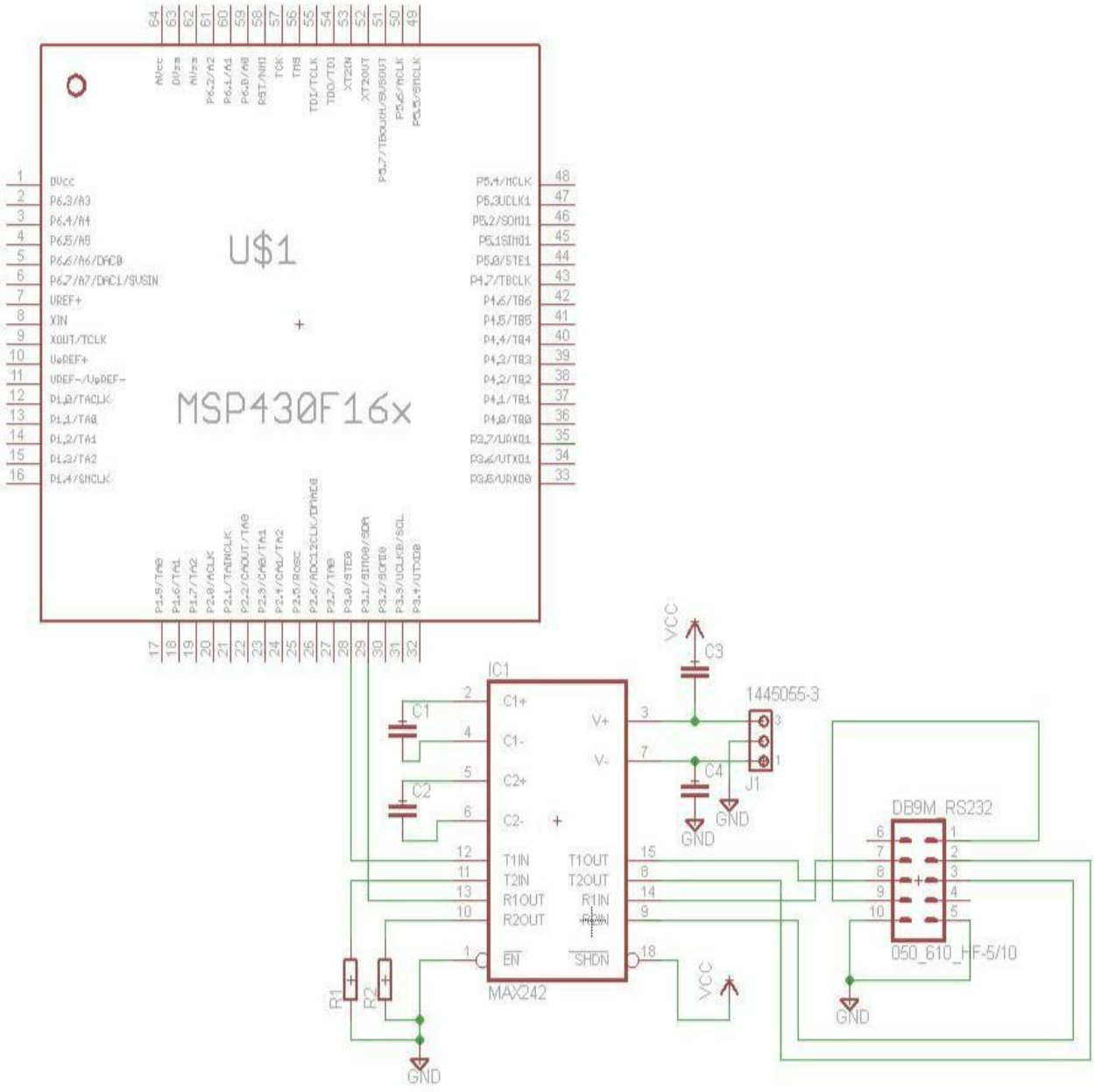
LABORATORIUM TM 5

Sobolewski Konrad , Boczek Bartłomiej

1.Cel laboratorium

Zadanie polegało na zaprojektowaniu oraz realizacji wyświetlania napisu przesuwanego, o zmiennym kierunku, na terminalu na komputerze PC, sterowanym przez port szeregowy z kontrolera MSP430.

2.Rysunek układu



3.Kod programu

```
#include <msp430.h>
#include <stdint.h>
#include <string.h>

//rozmiar wyswietlacza
#define size_display 17

//tablica na napis
char tab[]={"Lubie Nalesniki "};

//zmienna przechowujaca rozmiar tablicy
int size_tab = 0;

//tablica na wyswietlacz
char display[size_display+1];

//kierunek przesuwania napisu
unsigned short int right    = 1;

//flaga informujaca czy zaczac przesuwanie
unsigned short int startShift= 0;

const short int refresh = 64; //  32768Hz

//licznik do odmierzania czasu
int counter=0;

char temp;
char receiver;
unsigned short int keypressed=0;
int attempts=0;

void init()
{
    size_tab=0;
    //obliczanie rozmiaru tablicy
    while (tab[size_tab] != '\0')
    {
        ++size_tab;
    }

    int i=0;
    for(i=0;i<size_display;i++)
    {
        display[i]=tab[i];
    }
    display[size_display]='\r'; //na koncu wyswietlacza zawsze powrot karetki

    UCTL0 |= SWRST; //reset softwarowy


    TACTL = MC_1 | ID_0 | TASSEL_1; //clock init ACLK, mode Up, no divider.
```

```
53    BCSCTL1 &= ~XTS; //Basic Clock odule Operation, bardzo nieskie złużycie energii, 'low
    frequency' ACLK
54    TACCR0=refresh;  //wlaczenie timera, poprzez nadanie wartosci, timer liczy do tej wartosci
55
56    //Basic clock system control 2
57    BCSCTL2 |= SELS;// Korzystamy z zewnetrznego kwarcu jako zrodla taktowania, z SMCLK
58
59    P3SEL |= 0x30; //ustawiamy pierwsza i druga nozke jako Rx i Tx // P3.4 UTXD0 oraz P3.5 URDX0
60
61    //Module Enable Register
62    ME1 |= UTXE0 + URXE0;// Wlaczamy modul do wysylania danych i odbierania w UART
63
64    //  Parity Select + Parity Enable + character lenght + stop bit select
65    UCTL0 |= PEV + PENA + CHAR + SPB;// Ustawiamy bit parzystosci, dlugosc danych 8 bitow, i 2
    bity stopu
66
67
68    // Ustawienia dla 115200Hz (dla UART: max 115200 hz)
69    UTCTL0 |= SSEL1; //typowo 1,048,576-Hz SMCLK. --> LPM1
70    UBR00 = 0x40; // pierwszy dzielnik czestotliwosci
71    UBR10 = 0x00; // drugi dzielnik
72    UMCTL0 = 0x00;
73    UCTL0 &= ~SWRST; // wylaczenie software reset
74
75    TACCTL0 |= CCIE; //clock interrupts enabled
76    IE1 |= URXIE0; //przerwania wlaczone dla odbierania/dla wysylania
77    IE1 |= UTXIE0;
78    _BIS_SR(LPM1_bits + GIE); //deep sleep with smclk running + interrupts enabled
79  }
80
81  void shift()
82  {
83    int i = 0;
84    if(right)
85       {
86          temp=tab[size_tab-1];
87          for(i=size_tab-2;i>-1;i--)
88          {
89             tab[i+1]=tab[i];
90          }
91          tab[0]=temp;
92       }
93       else if(!right)
94       {
95          temp=tab[0];
96          for(i=1;i<size_tab;i++)
97          {
98            tab[i-1]=tab[i];
99          }
100         tab[size_tab-1]=temp;
101      }
102   for(i=0;i<size_display;i++)
103      display[i]=tab[i];
104   display[size_display]='\r';
```

```
}

int main(void)
{
    WDTCTL = WDTPW | WDTHOLD;    // kill the watchdog
    init();

    while (1)
    {
        if(keypressed && right) // przesun w lewo
        {
                right=0;
                keypressed=0;
        }
        else if(keypressed && !right) // w prawo
        {
                right=1;
                keypressed=0;
        }
        if(counter==512)  // jesli 1 sekunda minela
        {
            startShift=1; //moge przesunac
            counter=0;

        }
        if(startShift==1)
        {
            startShift=0;
            shift();//...to przesun napis
            IE1 |= UTXIE0;
        }
        _BIS_SR(LPM1_bits + GIE);
    }
    return 0;
}

#pragma vector=TIMERA0_VECTOR
__interrupt void timerA_int(void)
{
    counter++;//liczymy ile razy odswiezylismy

    _BIC_SR_IRQ(LPM1_bits);
}

#pragma vector=USART0TX_VECTOR
__interrupt void usart0_tx(void)
{
    TXBUF0 = display[attempts];
    attempts++;

    if(attempts>size_display){
        attempts=0;
        IE1 &= ~UTXIE0;   //If UTXIE0 is set, the UTXIFG0 flag will not trigger a transfer
    }
}
```

```
160
161 #pragma vector=USART0RX_VECTOR
162
163 __interrupt void usart0_rx (void)
164 {
165 // przerwanie od ukladu UART, odbieramy znak
166    receiver = (char)RXBUF0; //Receiver Buffer
167    if('q'==receiver)
168        keypressed=1;
169
170    _BIC_SR_IRQ(LPM1_bits);
171 }
```

4.Opis

Założyliśmy , że program po starcie przesuwa napis w prawą stronę ustawiając odpowiednią flagę. Posiadamy 2 tablice , tab na słowo oraz display do trzymania znaków wyświetlanych . Funkcja shift przesuwa znaki tab. Funkcja display trzyma określoną ilość pierwszych znaków tablicy tab. Przesunięcie tablicy display do PuTTy odbywa się co sekundę . Wtedy to przerwania transmitera są uaktywnione i wyłączamy je po przesłaniu wszystkich znaku tablicy display.Układ po wykonaniu jednej pętli w mainie jest usypiany. Wybudzają go przerwania związane z timerem oraz z receiverem. Układ udało się zrealizować z prędkością transmisji wynosząca 115 200 kHz.