LABORATORIUM TM 5

Sobolewski Konrad , Boczek Bartłomiej

1. Cel laboratorium

Zadanie polegało na cyklicznym przesuwaniu stałego napisu na wyświetlaczu 8-pozycyjnym - po naciśnięciu przycisku zmieniany jest kierunek ruchu.

2. Kod

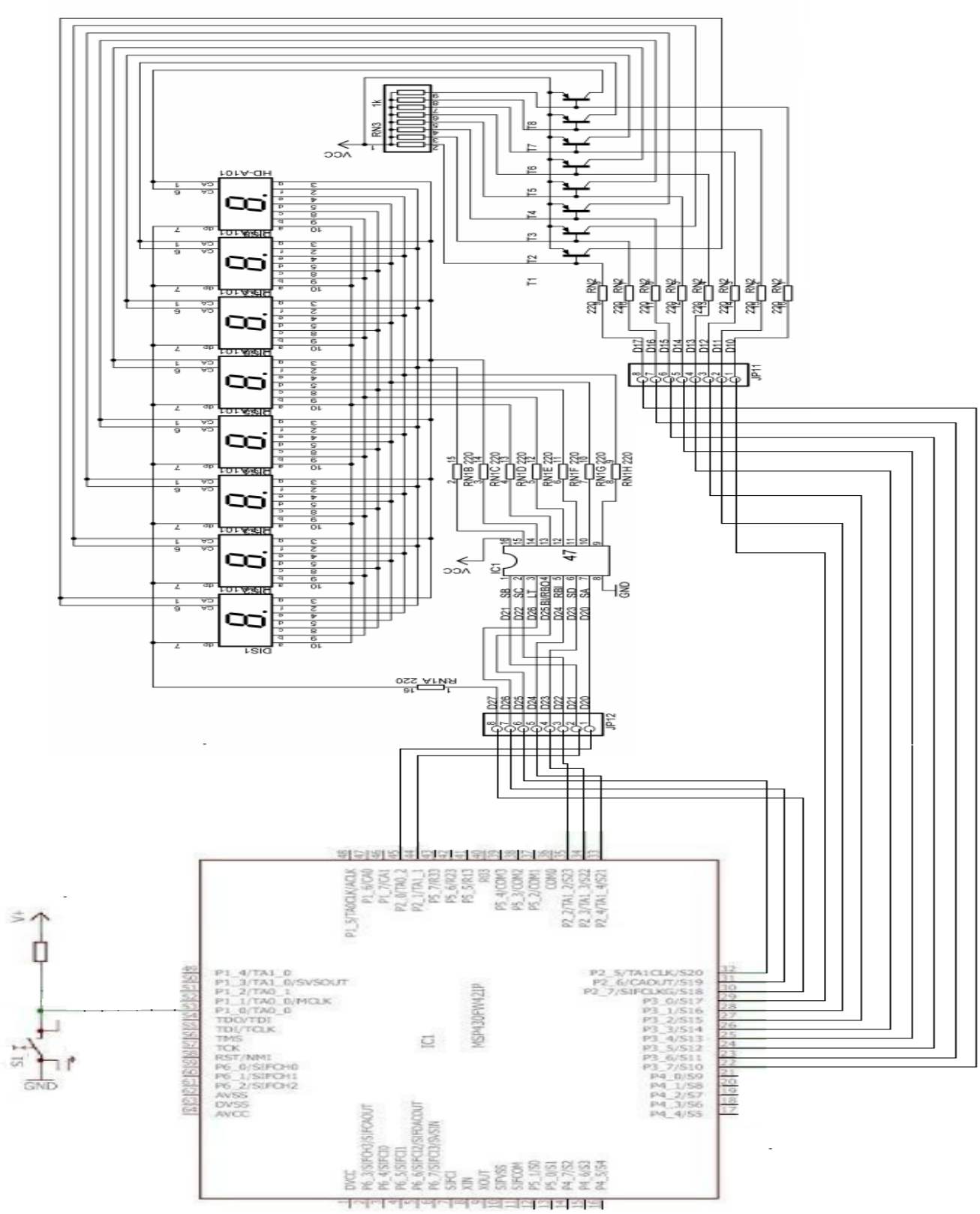
```
#include <msp430.h>
#include <stdint.h>
#define DISPLAY_SIZE 8
#define CZESTOTLIWOSC 0.5
unsigned short int tab[] = {1,2,3,4,5,0,0,0,7,8,9}; //tablica na dane

int start_poz = 0; //okresla od ktorej pozycji w tablicy z danymi zaczac wyswietlanie

const int tab_size = sizeof(tab)/sizeof(tab[0]); //rozmiar tablicy na dane
unsigned short int display[DISPLAY_SIZE]; //tablica reprezentujaca wyswietlacz
unsigned short int right = 1; //flaga ->przesuwamy w prawo czu w lewo
unsigned short int wiekszy_disp = 0; //flaga mowi czy rozmiar wyswietlacza jest wiekszy od napisu
short int direction =0x01; // przelacznik od zmiany kierunku
unsigned short int startShift= 0; // czy przesuwac
const unsigned short int  refresh = 64; //  32678Hz
short int position=0;  //indeks tablicy do wyswietlenia
int counter=0;
int counter_odszum = 0;
int odszum = 0;
init()
{
    unsigned int i = 0;
    //inicjalizacja wyswietlacza
    if (tab_size>=DISPLAY_SIZE)
    {
        for (i=0; i<DISPLAY_SIZE; ++i)
        {
            display[i] = tab[i];
        }
    }
    else
    {
        for (i=0; i<tab_size; ++i)
        {
            display[i] = tab[i];
        }
        for (i; i<DISPLAY_SIZE; ++i)
        {
            display[i] = 0; //lub puste
        }
    }
    for (i=0;i<tab_size;++i)
```

```
43      {
44          tab[i]+=224; //zeby sie dobrze wyswietlalo
45      }
46      if (DISPLAY_SIZE > tab_size)
47      {
48        wiekszy_disp = 1;
49      }
50      else
51      {
52          wiekszy_disp = 0;
53      }
54
55      TACTL = MC_1 | ID_0 | TASSEL_1; //mode up ,bez dzielnika,init clock;
56      BCSCTL1 &= ~ XTS; // wolny tryb ACLK
57
58      TACCTL0 |= CCIE; //wlacz przerwania zegara
59      TACCR0 = refresh; //zliczaj 1 sekunde
60
61      P1DIR=0x00 ; // kieruenk wejsciowy
62      P1IE |= direction;  // wlacz przerwania dla zmiany kierunku...
63      P1IES |= BIT0;  // .. po rosnacym zboczu
64      P1IFG &= ~direction;  //  wyczysc rejest flag
65
66      P3DIR = 0xff; // pozycja to highlight
67      P2DIR = 0xff; // pozycja do wyswietlenia
68
69      _BIS_SR(LPM3_bits + GIE); // idz spac i wlacz przerwania
70  }
71
72  void refresh_display()
73  {
74      P4OUT |= 0xFF; // wylacz wyswietlacz
75      if(position>7)
76      {
77          position=0;
78      }
79      if(display[position]>0)
80      {
81          P3OUT &= ~(1<<position); // wybierz nowa pozycje do wyswietlanie !!!
82          P2OUT = display[position]; // wyswietl to na pozycji wybranej powyzej
83      }
84      position++;
85  }
86
87  void shift_wiekszy_display()
88  {
89        int i = 0, j=0,k = 0;
90      //zerowanie wyswietlacza
91      for (i=0;i<DISPLAY_SIZE;++i){
92          display[i] = 0;
93      }
94      if (right)
95      {
96        --start_poz;
97      }
```

```
98      else
99      {
100        ++start_poz;
101     }
102     if (start_poz > DISPLAY_SIZE-1)
103     {
104       start_poz = 0;
105     }
106     else if(start_poz < 0)
107     {
108       start_poz = DISPLAY_SIZE-1;
109     }
110     for (i = start_poz, j=0; i<DISPLAY_SIZE && j<tab_size;++i,++j)
111     {
112       display[i] = tab[j];
113     }
114
115     k = j;
116
117     for (i = 0,k; i<DISPLAY_SIZE-j && k<tab_size; ++k,++i)
118     {
119       display[i] = tab[k];
120     }
121 }
122
123
124 void shift ()
125 {
126       int i = 0, j=0,k = 0;
127     if (right)
128     {
129       --start_poz;
130     }
131     else
132     {
133       ++start_poz;
134     }
135     if (start_poz > tab_size-1)
136     {
137       start_poz = 0;
138     }
139     else if(start_poz < 0)
140     {
141       start_poz = tab_size-1;
142     }
143     for (i = start_poz, j=0; i<tab_size && j<DISPLAY_SIZE;++i,++j)
144     {
145       display[j] = tab[i];
146     }
147     k = j;
148
149     for (i = 0,k; i<tab_size-j && k<DISPLAY_SIZE; ++k,++i)
150     {
151       display[k] = tab[i];
152     }
```

```
153 }
154
155 main(void)
156 {
157    WDTCTL = WDTPW | WDTHOLD ; // wylacz watchdoga
158    init();
159    while(1)
160    {
161       if(odszum && counter_odszum > 200)
162       {
163          odszum=0;
164          if(P1IN & right) // przesun w lewo //mozliwe ze trzeba zrobic (P1IN & BIT0) & right
165          {
166             right=0;
167          }
168          else if(P1IN & !right) // w prawo
169          {
170             right=1;
171          }
172          P1IFG &= ~BIT0; // wyczysc flagi
173       }
174
175       counter++;//liczymy ile razy odswiezylismy
176       if(counter==512*CZESTOTLIWOSC)  // jesli 1 sekunda minela
177       {
178             startShift=1; //moge przesunac
179             counter=0;
180       }
181       if(startShift)//jesli 1 sekunda..
182       {
183          startShift=0;
184          //...to przesun napis
185
186          if (wiekszy_disp)
187           {
188             shift_wiekszy_display();
189          }
190            else
191          {
192             shift();
193          }
194       }
195       _BIS_SR(LPM3_bits); // idz spac
196    }
197 }
198
199 #pragma vector=TIMERA0_VECTOR
200 __interrupt void timerA_int(void)  // wchodzimy tu co 512 razy/sekunde
201 {
202    refresh_display(); // odswiez ekran
203
204    if (P1IN) //inkrementuj tylko jezeli przycisk jest wcisniety
205       ++counter_odszum; //inkrementacja licznika do odszumiania
206    else
207       counter_odszum = 0; //jezeli nie wcisniety to znaczy ze trzeba zaczac liczyc od nowa
```

```
208     LPM3_EXIT;  // obudz sie
209 }
210 
211 #pragma vector=PORT1_VECTOR
212 __interrupt void switches_int(void) // wchodzimy tu jak zmienimy stan przelacznika
213 {
214     counter_odszum=0;
215     odszum=1;
216     P1IFG &= ~BIT0; // czyszczenie flagi przerwan
217 }
```


RN3 1k
VCC
T8
T7
T6
T5
T4
T3
T2
T1
RN2 220
D17 D16 D15 D14 D13 D12 D11 D10
JP11
HD-A101
DIS1
RN1A 220
RN1B 220
RN1C 220
RN1D 220
RN1E 220
RN1F 220
RN1G 220
RN1H 220
IC1
47
GND
D21 SB
D22 SC
D26 LT
D25 BI/RBO
D24 RBI
D23 SD
D20 SA
D27 D26 D25 D24 D23 D22 D21 D20
JP12
V+
S1
GND
IC1
MSP430FW42IP
P1_4/TA1_0
P1_3/TA1_0/SVSOUT
P1_2/TA0_1
P1_1/TA0_0/MCLK
P1_0/TA0_0
TDO/TDI
TDI/TCLK
TMS
TCK
RST/NMI
P6_0/SIFCH0
P6_1/SIFCH1
P6_2/SIFCH2
AVSS
DVSS
AVCC
P2_5/TA1CLK/S20
P2_6/CAOUT/S19
P2_7/SIFCLKG/S18
P3_0/S17
P3_1/S16
P3_2/S15
P3_3/S14
P3_4/S13
P3_5/S12
P3_6/S11
P3_7/S10
P4_0/S9
P4_1/S8
P4_2/S7
P4_3/S6
P4_4/S5