Приложение А

Листинг программы

Файл main.c:

```
#include "main.h"
unsigned int mode;
unsigned char data;
int i=0;
int q=0;
int n=0;

void port_ini(void)
{
    DDRA=0x0F;
    PORTD=0x00;
    DDRD=0x20;
    PORTB=0xFF;
    DDRB=0x00;
    PORTC=0xFF;
    DDRC=0x00;
}

ISR(USART_RX_vect)
{
    data = UDR;
    if ((data=='1')||(data=='2')||(data=='3')||(data=='4')||(data=='5')||(data=='6')||(data=='7')||(data=='8')||(data=='9')||(data=='0')||(data==',')||(data=='p')||(data=='d')){
        if (mode==1)
        {
            data = (data<<4);
            data = (data>>4);
            Play(data);
        } else {
            if (i == 196)
            {
                USART_Send_Str("error!");
                USART_Send_Str("limit reached");
                goto m2;
            }
            if (data=='p')
            {
                if (n!=3)
                {
                    USART_Send_Str("error!");
                    USART_Send_Str("last note is too short");
                    goto m2;
                }
                i++;
                buffermel[i]=',';

                SaveBuff(i);
                n=0;
                for (int ii=0;ii<i;ii++)
                {
                    buffermel[ii]=NULL;
                }
                i=0;
                goto m2;
            }
            if (data=='d')
            {
                i--;
                if (n==0) {n=3;}else{n--;}
                goto m2;
            }
            if (n==3)
            {
                if (data!=',')
                {
                    USART_Send_Str("error!");
                    USART_Send_Str("expected ','");
                    goto m2;
                }
            }
            if (data==',')
            {
                if (n<3)
                {
                    USART_Send_Str("error!");
                    USART_Send_Str("correct note is 3 chr long");
                    goto m2;
                }
                n=0;
                goto m3;
            }

            n++;
m3:         buffermel[i]=data;
            i++;
m2:         USART_Send_Str("melody:");
            for (q=0; q<i; q++){USART_Send_Char(buffermel[q]);}
        }
    } else
    {
        USART_Send_Str("error!");
        USART_Send_Str("Put numbers 1..7");
    }
}

int main(void)
{
    unsigned char count;
    unsigned char temp;
    unsigned char countc;
    unsigned char tempc;

    port_ini();
    USART_ini(12);
    sei();
    mode = 0;
    PORTA=mode+1;
    while (1)
    {
        m1: tempc=PINC;
        for (countc=0; countc<4; countc++)
```

```
{
    if ((tempc&1)==0) {
        if (countc<2){
            mode = countc;
            PORTA=mode+1;
        }
        if (countc==2)
        {
            if (mode==0){ChangeTempo(0);PORTA=mode+1;}
            else {ChangeOctave(0);PORTA=mode+1;}
        }
        if (countc==3)
        {
            if (mode==0){ChangeTempo(1);PORTA=mode+1;}
            else {ChangeOctave(1);PORTA=mode+1;}
        }
    }
    tempc >>= 1;
}
temp=PINB;

for (count=0; count<8; count++)
{
    if ((temp&1)==0) {
        if (mode==1) {
            count++;
            Play(count);
            goto m1;
        } else {
            switch (count)
            {
                case 0:
                {
                    PlayMel(&mel0[0]);
                }
                break;
                case 1:
                {
                    PlayMel(&mel1[0]);
                }
                break;
                case 2:
                {
                    PlayMel(&mel2[0]);
                }
                break;
                case 3:
                {
                    PlayMel(&mel3[0]);
                }
                break;
                case 4:
                {
                    PlayMel(&mel4[0]);
                }
                break;
```

```
                        case 5:
                        {
                            PlayMel(&mel5[0]);
                        }
                        break;
                        case 6:
                        {
                            PlayMel(&mel6[0]);
                        }
                        break;
                        case 7:
                        {
                            PlayMel(&mel7[0]);
                        }
                        break;
                    }
                    goto m1;
                    }
                }
            temp >>= 1;
            }
        }
        TCCR1A=0x00;
    }
```

Файл main.h:

```
#ifndef MAIN_H_
#define MAIN_H_

#define F_CPU 1000000UL

#include <avr/io.h>
#include <avr/interrupt.h>
#include <util/delay.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <avr/pgmspace.h>

#include "usart.h"
#include "Play.h"

#endif /* MAIN_H_ */
```

Файл play.h:

```
 #ifndef PLAY_H_
 #define PLAY_H_

 #include "main.h"
 #include "play.c"

void Play(int nota);
void PlayMel(unsigned char *mel);
void ChangeTempo(int t);
void ChangeOctave(int o);
void SaveBuff(i);

#endif //PLAY_H_
```

Файл play.c:

```
#include "Play.h"

const unsigned int tabz[]  = {4,8,16,32,64,128,256};
const unsigned char mel0[] = {132, 141, 141, 139, 141, 139, 141, 137, 132, 132, 132, 141, 141, 142, 139, 176, 128,
144, 146, 146, 154, 154, 153, 151, 149, 144, 153, 153, 151, 153, 181, 128, 96, 255};
const unsigned char mel1[] = {74, 109, 128, 64, 77, 106, 128, 64, 79, 77, 79, 77, 79, 77, 79, 77, 79, 113, 128, 64,
255};
const unsigned char mel2[] = {144, 143, 141, 136, 144, 143, 141, 136, 144, 143, 141, 136, 146, 144, 143, 144, 255};
const unsigned char mel3[] = {75, 64, 79, 32, 43, 32, 43, 47, 32, 43, 32, 53, 32, 75, 64, 82, 32, 43, 32, 43, 51, 32, 50,
46, 32, 43, 32, 50, 32, 55, 32, 43, 42, 32, 41, 38, 32, 45, 75, 255};
const unsigned int PROGMEM tabkd[] = {0, 4748, 4480, 4228, 3992, 3768, 3556, 3356, 3168, 2990, 2822, 2664,
2514, 2374, 2240, 2114, 1996, 1884, 1778, 1678, 1584, 1495, 1411, 1332, 1257, 1187, 1120, 1057, 998, 942, 889,
839, 792};
const unsigned int PROGMEM tabkd1[] = {0, 3768, 3356, 2990, 2822, 2514, 2240, 1996, 1884, 1678,  1495, 1411,
1257, 1120, 998, 942, 839};
unsigned char mel4[20];
unsigned char mel5[20];
unsigned char mel6[20];
unsigned char mel7[20];

char buffermel[100];

int buf=0;
unsigned int tempo=2;
unsigned int oct=0;

void ChangeTempo(int t)
{

        if (t == 0){
                if (tempo<4) {tempo=tempo+1; PORTA=0x08;_delay_ms(500);}
                }
        if (t == 1){
                if (tempo>1){tempo=tempo-1;PORTA=0x04;_delay_ms(500);}
                }
}

void ChangeOctave(int o)
{
        if (o == 0){
                if (oct<7) {oct=oct+7; PORTA=0x08;_delay_ms(500);}
        }
        if (o == 1){
                if (oct>6) {oct=oct-7;PORTA=0x04;_delay_ms(500);}
        }
}

void SaveBuff(int g)
```

```
{
    int b;
    int z;
    int nt=0;
    int tempp=0;
    unsigned char bufferstr[3];
    char *uk=&bufferstr[0];
    USART_Send_Char((unsigned char) g);
    g=g+3;
    for (b=0; b<=g; b++)
    {
        if (buffermel[b]==',')
        {
            b=b-3;

            for (z=0; z<3; z++)
            {
                bufferstr[z]=buffermel[(b+z)];

            }
            b=b+4;
            tempp=atoi(uk);
            if (buf == 0){

                    mel4[nt] =(char) tempp;
                }
            if (buf == 1)
                {
                    mel5[nt] =(char) tempp;
                }
            if (buf == 2)
                {
                    mel6[nt] =(char) tempp;
                }
            if (buf == 3)
                {
                    mel7[nt] =(char) tempp;
                }
            nt++;
        }
    }

    switch (buf)
    {
        case 0:
        {
            mel4[nt]=0xFF;
            USART_Send_Str("melody is on PB4");
        }
        break;
        case 1:
        {
            mel5[nt]=0xFF;
            USART_Send_Str("melody is on PB5");
        }
        break;
        case 2:
        {
            mel6[nt]=0xFF;
            USART_Send_Str("melody is on PB6");
        }
        break;
        case 3:
        {
            mel7[nt]=0xFF;
            USART_Send_Str("melody is on PB7");
        }
        break;
    }

    buf++;
    if (buf==4)
    {
        buf=0;
    }

}

void Play(int nota)
{
    if (pgm_read_word_near(&tabkd1[nota]) != NULL)    {
        TCCR1A=0x00;
        TCCR1B=0x09;
        nota = nota + oct;
        OCR1A= (pgm_read_word_near(&tabkd1[nota]));
        TCCR1A=0x40;
        _delay_ms(200);
        TCCR1A=0x00;
    }
}


void PlayMel(unsigned char *mel)
{
    unsigned char fnota;
    unsigned char dnota;
    unsigned char *nota;

    TCCR1A=0x00;
    TCCR1B=0x09;



        m3:     nota = mel;
        m4:     if (PINB==0xFF) goto m2;
        if (*nota==0xFF) goto m3;
        fnota = (*nota)&0x1F;
        dnota = ((*nota)>>5)&0x07;
        if (fnota==0) goto m5;
        OCR1A = (pgm_read_word_near(&tabkd[fnota]));
        TCCR1A=0x40;
        m5:     for (int i = 0; i < (2*tabz[dnota]/tempo); i++)
        {
            _delay_ms(10);
        }
        TCCR1A=0x00;
        for (int i = 0; i < (2*tabz[0]/tempo); i++)
        {
            _delay_ms(10);
        };
        nota++;
        goto m4;
    m2:TCCR1A=0x00;
}
```

Файл usart.h:

```
#ifndef USART_H_
#define USART_H_

#include "main.h"
#include "usart.c"

void USART_ini(unsigned int speed);
void USART_Send_Char (unsigned char data);
void USART_Send_Str(char str[]);
#endif /* USART_H_ */
```

Файл usart.c:

```
#include "usart.h"
void USART_ini (unsigned int speed)
{
    UBRRH = (unsigned char) (speed>>8);
    UBRRL = (unsigned char) speed;
    UCSRB = (1<<RXEN)|(1<<TXEN); //включаем прием и передачу по usart
    UCSRB |= (1<<RXCIE); //разрешаем прерывания при приеме
    //UCSRA |= (1<<U2X); //удвоение частоты
    UCSRC = (1<<URSEL)|(0<<USBS)|(3<<UCSZ0); // обращаемся к регистру UCSRC (URSEL=1),
ассинхронный режим (UNSEL=0),
    // без контроля четности (UPM1=0 UPM0=0), 2стоп -бита (USBS=1), 8битовая посылка (UCSZ1=1 и
UCZ0=1)
}
void USART_Send_Char (unsigned char data)
{
    while(!(UCSRA&(1<<UDRE)));
    UDR = data; //начнем передавать данные, но только убедившись в том что буфер пуст
}
void USART_Send_Str(char str[])
{
    unsigned char i = 0;
    USART_Send_Char(0x0d);
    while (str[i]!='\0')
    {
        USART_Send_Char(str[i]);
        i++;
    }
    USART_Send_Char(0x0d);
}
```