

جهت خرید فایل word به سایت www.kandoo.cn.com مراجعه کنید
یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۶۶۴۱۲۶۰-۰۵۱۱ تماس حاصل نمایید

فصل اول

سخت افزار

۱-۱- سخت افزار

۱- میکرو کنترلر ATmega8

۲- رابط USB

۳- LCD جهت ارتباط با کاربر

۴- رله و راه انداز آن

۱-۱-۱- میکرو کنترلر :

در این پروژه از ATmega8 به عنوان کنترلر و مدار فرمان رله ها استفاده شده است . در عین حال برای ارتباط با کامپیوتر از طریق پورت USB نیز ، از این میکرو کنترلر استفاده شده است. این میکرو کنترلر از خانواده میکروکنترلر های AVR است .

پروتکل USB به صورت نرم افزاری در برنامه (Firmware) میکرو پیاده سازی شده است و می تواند با استاندارد USB 1.1 Low به تبادل اطلاعات بپردازد.

برنامه این میکرو بر اساس کامپایلر GCC نوشته شده است که برای کامپایل آن می توان از برنامه WinAvr استفاده کرد .

LCD بکار رفته در این پروژه کاراکتری و ۱۶*۲ می باشد که بصورت مستقیم توسط نرم افزار کامپیوتری ، متن یا دستورالعمل را گرفته و اجرا می کند .

در برنامه کامپیوتری برای ارتباط با پورت USB از درایور آماده libusb استفاده شده است .

برنامه کامپیوتری توسط زبان C++Builder تهیه شده است. این برنامه مستقیماً از میکروفن کامپیوتر صوت را دریافت کرده و پس از ذخیره در بافر داخلی شروع به تجزیه و تحلیل می کند. روال کار به این صورت است که از اطلاعات ورودی که در بازه زمان هستند تبدیل فوریه (FFT) گرفته شده تا به بازه

با نمونه های ذخیره شده مقایسه می کند .

۱-۱-۲- لخت افزار Firmware :

میکرو میتواند پنج دستورالعمل مختلف را از کامپیوتر دریافت نموده و اجرا نماید .

۱- ارسال یک بایت دستورالعمل به LCD : `cmdLCD_Command`

۲- ارسال یک بایت دیتا به LCD : `cmdLCD_Data`

۳- نمایش یک رشته متنی روی LCD : `cmdLCD_Str`

۴- روشن و خاموش کردن `cmdLCD_BL : BackLight`

۵- کنترل رله ها : `cmdSWITCH`

برای کنترل LCD سه تابع طراحی شده است :

`void LCD_Command(uchar command)`

`void LCD_Data(uchar data)`

`void LCD_Init(void)`

این توابع برای ارسال دستورالعمل و دیتا و همچنین تنظیمات اولیه استفاده می شوند .

یکی از توابعی که در بخش USB آن استفاده می شود `usbFunctionSetup` است که توسط آن به

درخواست رسیده شده از کامپیوتر پاسخ داده می شود . از تابع `usbFunctionWrite` جهت دریافت

اطلاعات از کامپیوتر استفاده می گردد . تابع دیگری که برای ارسال دیتا به کامپیوتر از آن استفاده می شود

`usbFunctionRead` نام دارد که در این پروژه از آن استفاده نگردیده است .

متن کامل این توابع را در زیر مشاهده می کنید :

```
PORTB &= 0b11000111;  
PORTC &= 0b11100000;  
PORTC |= (((command & 0b00000001) << 4) & 0b00010000);  
PORTC |= (((command & 0b00000010) << 2) & 0b00001000);  
PORTC |= (((command & 0b00000100) >> 0) & 0b00000100);  
PORTC |= (((command & 0b00001000) >> 2) & 0b00000010);  
PORTC |= (((command & 0b00010000) >> 4) & 0b00000001);  
PORTB |= (((command & 0b00100000) >> 0) & 0b00100000);  
PORTB |= (((command & 0b01000000) >> 2) & 0b00010000);  
PORTB |= (((command & 0b10000000) >> 4) & 0b00001000);  
PORTD &= ~(1 << 6);  
PORTC |= (1 << 5);  
_delay_ms(10);  
PORTC &= ~(1 << 5);  
}
```

```
void LCD_Data(uchar data){
```

```
PORTB &= 0b11000111;  
PORTC &= 0b11100000;  
PORTC |= (((data & 0b00000001) << 4) & 0b00010000);  
PORTC |= (((data & 0b00000010) << 2) & 0b00001000);  
PORTC |= (((data & 0b00000100) >> 0) & 0b00000100);  
PORTC |= (((data & 0b00001000) >> 2) & 0b00000010);  
PORTC |= (((data & 0b00010000) >> 4) & 0b00000001);  
PORTB |= (((data & 0b00100000) >> 0) & 0b00100000);
```

```
PORTB |= (((data & 0b10000000) >> 4) & 0b00001000);
```

```
PORTD |= (1 << 6);
```

```
PORTC |= (1 << 5);
```

```
_delay_ms(10);
```

```
PORTC &= ~(1 << 5);
```

```
}
```

```
void LCD_Init(void){
```

```
LCD_Command(0x38);
```

```
_delay_ms(100);
```

```
LCD_Command(0x0C);
```

```
_delay_ms(100);
```

```
LCD_Command(0x01);
```

```
_delay_ms(100);
```

```
}
```

```
uchar usbFunctionSetup(uchar data[8]) {
```

```
    uchar len = 0;
```

```
    if(data[1] == cmdLCD_Command){
```

```
LCD_Command(data[2]);
```

```
replyBuffer[0] = cmdOK;
```

```
len = 1;
```

```
    } else if (data[1] == cmdLCD_Data) {
```

```
replyBuffer[0] = cmdOK;
```



```
    } else if (data[1] == cmdLCD_Str) {  
len = 0xff;  
    } else if (data[1] == cmdLCD_BL) {  
if(data[2] != 0)  
PORTD |= (1 << 7);  
else  
PORTD &= ~(1 << 7);  
replyBuffer[0] = cmdOK;  
len = 1;  
    } else if (data[1] == cmdSWITCH) {  
sw = data[4];  
PORTB |= 0b00000100;  
PORTD |= 0b00111010;  
PORTB &= (((sw | 0b1111110) << 2) | 0b11111011);  
PORTD &= (((sw | 0b1111101) << 5) | 0b11011111);  
PORTD &= (((sw | 0b11111011) << 4) | 0b11101111);  
PORTD &= (((sw | 0b11110111) << 3) | 0b11110111);  
PORTD &= (((sw | 0b11101111) << 2) | 0b11111011);  
replyBuffer[0] = cmdOK;  
len = 1;  
    }  
    usbMsgPtr = replyBuffer;  
return len;  
}
```

جهت خرید فایل word به سایت www.kandoocn.com مراجعه کنید

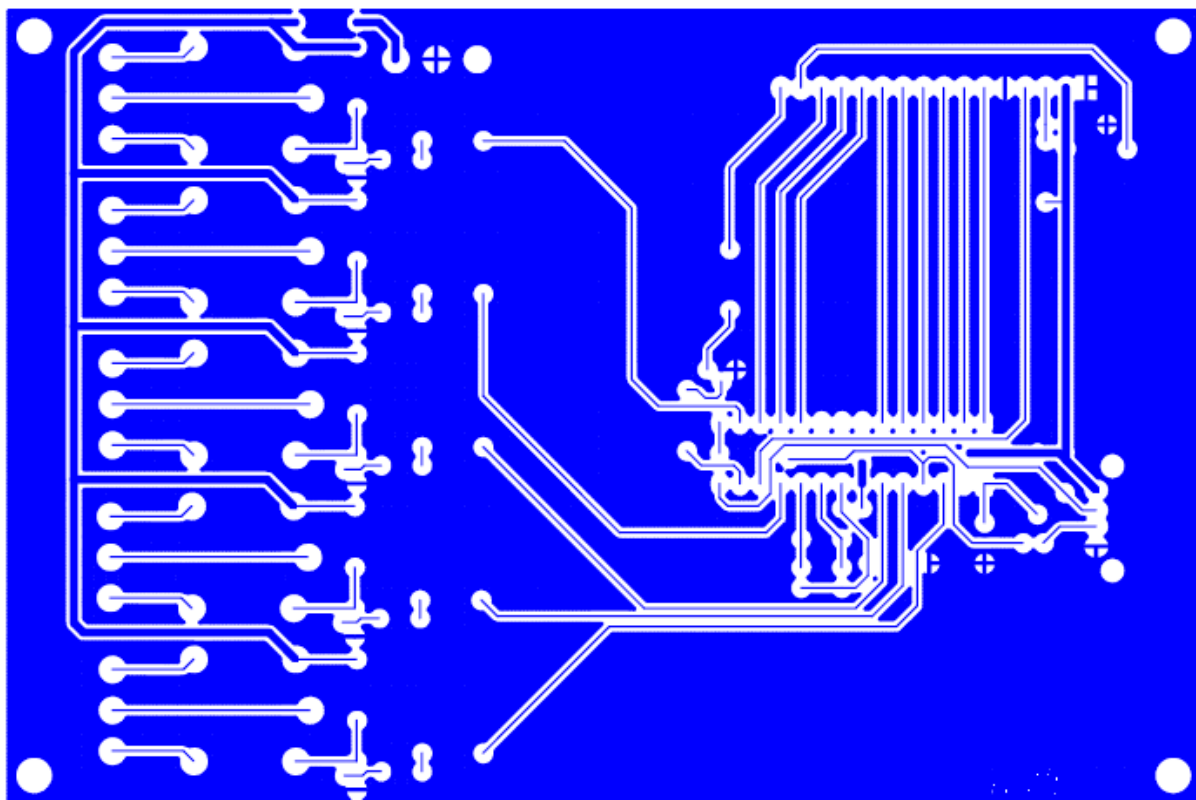
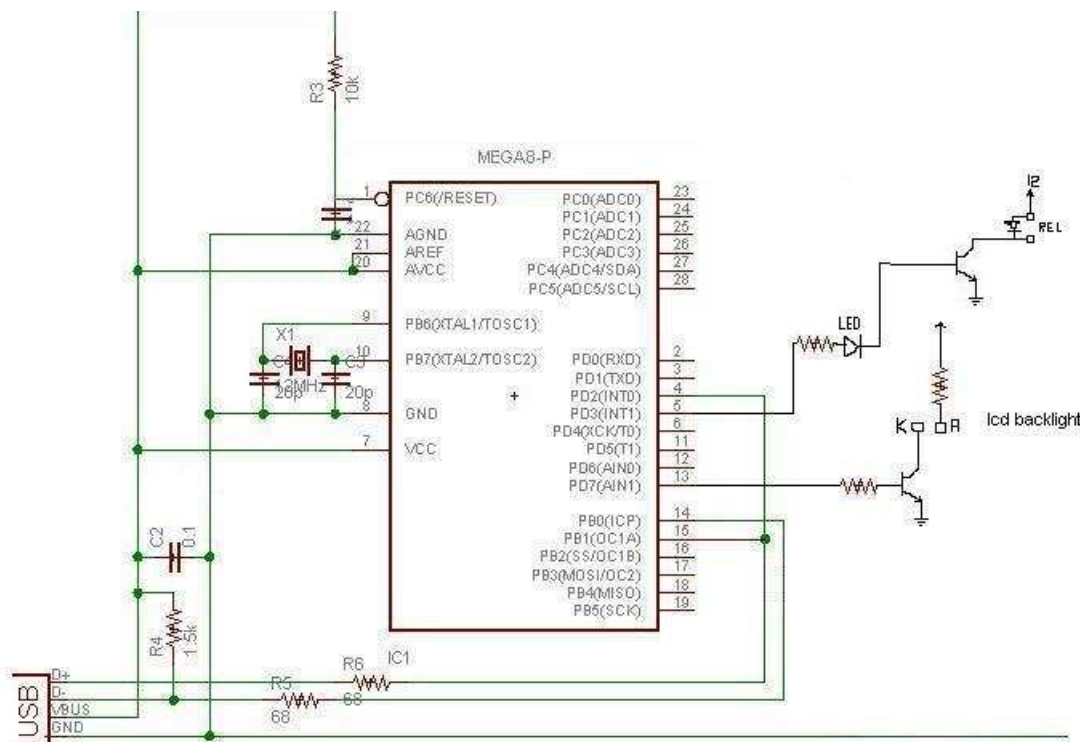
یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۶۶۴۱۲۶۰-۰۵۱۱ تماس حاصل نمایید

```
return 0;  
}
```

```
uchar usbFunctionWrite(uchar *data, uchar len) {  
    uchar i;  
    for (i = 0; i < len; i++) {  
        LCD_Data(data[i]);  
    }  
    return 0;  
}
```

طرح شماتیک و نقشه پشت برد مدار بصورت زیر خواهد بود:

جهت خرید فایل word به سایت www.kandoo.cn.com مراجعه کنید
 یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۶۶۴۱۲۶۰-۵۱۱ تماس حاصل نمایید



جهت خرید فایل word به سایت www.kandooen.com مراجعه کنید
یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۶۶۴۱۲۶۰-۰۵۱۱ تماس حاصل نمایید

فصل دوم

نرم افزار

Software

۱-۲- نرم افزار:

در این برنامه برای ارتباط از طریق پورت USB از درایور libusb-win32 استفاده شده است . برای

استفاده از آن باید یک فایل کتابخانه ای به نام LIBUSB.lib را به پروژه اضافه کرده و از توابع

libusb0.dll استفاده نماییم.

توابع استفاده شده در این پروژه به این شرح می باشند :

```
void usb_init(void);
```

```
int usb_find_busses(void);
```

```
int usb_find_devices(void);
```

```
usb_dev_handle *usb_open(struct *usb_device dev);
```

```
int usb_close(usb_dev_handle *dev);
```

```
int usb_control_msg(usb_dev_handle *dev, int requesttype, int request, int
```

```
value, int index, char *bytes, int size, int timeout);
```

برای نقل و انتقال اطلاعات و دستورالعمل از تابع زیر استفاده می شود :

```
static int usbtransmit(unsigned char receive, unsigned char functionid,
```

```
unsigned char send[4], unsigned char * buffer, int
```

```
buffersize)
```

```
{
```

```
int nbytes;
```

```
nbytes = usb_control_msg(usbhandle,
```

```
USB_TYPE_VENDOR | USB_RECIP_DEVICE | (receive
```

```
<< 7),
```

```
functionid,
```

```
(send[1] << 8) | send[0],
```

```
(send[3] << 8) | send[2],
```

```
buffer, buffersize,
```

```
if(nbytes < 0){  
    MessageDlg("ControlMessage Error." , mtError, TMsgDlgButtons() <<  
mbOK, 0);  
    exit(1);  
}  
  
return nbytes;  
}
```

این تابع در واقع پایه همه توابعی است که در برنامه آمده است.

سایر توابع مورد نیاز به این شرح است :

unsigned char USB_LCD_Command(unsigned char byte)

```
{  
    unsigned char res[4];  
    unsigned char cmd[4];  
    cmd[0]=byte;  
    usbtransmit(1, cmdLCD_Command,cmd, res,sizeof(res));  
    return res[0];  
}
```

unsigned char USB_LCD_Data(unsigned char byte)

```
{  
    unsigned char res[4];  
    unsigned char cmd[4];  
    cmd[0]=byte;  
    usbtransmit(1, cmdLCD_Data,cmd, res,sizeof(res));  
    return res[0];  
}
```

```
unsigned char USB_LCD_Str(unsigned char *data,unsigned char len)
```

```
{  
    unsigned char cmd[4];  
    usbtransmit(0, cmdLCD_Str, cmd, data, len);  
    return 0;  
}
```

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

```
unsigned char USB_LCD_BL(unsigned char byte)
```

```
{  
    unsigned char res[4];  
    unsigned char cmd[4];  
    cmd[0]=byte;  
    usbtransmit(1, cmdLCD_BL,cmd, res,sizeof(res));  
    return res[0];  
}
```

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

```
unsigned char USB_SWITCH(unsigned char byte)
```

```
{  
    unsigned char res[4];  
    unsigned char cmd[4];  
    cmd[0]=byte;  
    usbtransmit(1, cmdSWITCH,cmd, res,sizeof(res));  
    return res[0];  
}
```

اولین آرگومان تابع `usbtransmit` بیانگر ورودی/کنترلی و یا خروجی بودن فراخوانی آن را دارد که در

توابع فوق فقط تابع `USB_LCD_Str` خروجی است و بقیه تابع کنترلی هستند.

جهت خرید فایل word به سایت www.kandoocn.com مراجعه کنید

یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۶۶۴۱۲۶۰-۰۵۱۱ تماس حاصل نمایید

برای پرسش سوالات در پیروزی از تویچ www.kandoocn.com یا در این پرورش سوالات سوالات

نمونه برداری ۱۱۰۲۵ و بصورت ۸ بیتی استفاده شده است . طول هر کلمه حد اکثر ۲ ثانیه در نظر گرفته

شده که برای ذخیره آن ۲*۱۱۰۲۵ بایت بافر کافیست .

توابع زیر از port.dll مورد استفاده قرار گرفته اند :

SOUNDIN

SOUNDOUT

SOUNDGETRATE

SOUNDSETRATE

SOUNDGETBYTES

SOUNDSETBYTES

SOUNDCAPIN

SOUNDCAPOUT


```
#include <vcl.h>
#pragma hdrstop
```

```
#include <stdio.h>
#include <math.h>
```

```
#define USBDEV_VENDOR 0x03eb
#define USBDEV_PRODUCT 0x7a53
```

```
#define cmdLCD_Command 0
#define cmdLCD_Data 1
#define cmdLCD_Str 2
#define cmdLCD_BL 3
#define cmdSWITCH 4
#define cmdOK 5
```

```
#include "usb.h"
```

```
typedef UINT (CALLBACK* LPFNDLLFUNC1)(unsigned char *,DWORD);(
typedef UINT (CALLBACK* LPFNDLLFUNC2)(unsigned char *,DWORD);(
typedef UINT (CALLBACK* LPFNDLLFUNC3);()(
typedef UINT (CALLBACK* LPFNDLLFUNC4)(unsigned long);(
typedef UINT (CALLBACK* LPFNDLLFUNC5);()(
typedef UINT (CALLBACK* LPFNDLLFUNC6)(unsigned long);(
typedef UINT (CALLBACK* LPFNDLLFUNC7);()(
typedef UINT (CALLBACK* LPFNDLLFUNC8);()(
HINSTANCE hDLL;
LPFNDLLFUNC1 soundin;
LPFNDLLFUNC2 soundout;
LPFNDLLFUNC3 getrate;
LPFNDLLFUNC4 setrate;
LPFNDLLFUNC5 getbytes;
LPFNDLLFUNC6 setbytes;
LPFNDLLFUNC7 capin;
LPFNDLLFUNC8 capout;
FILE * set;
float l=0;
bool _usb_open=false;
```

```
#include "Unit1.h"
```

```
-----//
usb_dev_handle *usbhandle;
struct usb_bus *usb_busses;
```

```
static int usbtransmit(unsigned char receive, unsigned char functionid,  
                      unsigned char send[4], unsigned char * buffer, int buffersize(  
}  
int nbytes;  
nbytes = usb_control_msg(usbhandle,  
                          USB_TYPE_VENDOR | USB_RECIP_DEVICE | (receive << 7, (  
                          functionid,  
                          ) send[1] << 8) | send[0, (  
                          ) send[3] << 8) | send[2, (  
                          buffer, buffersize,  
                          );(۰۰۰۰  
if(nbytes < 0) (  
    MessageDlg("ControlMessage Error." , mtError, TMsgDlgButtons() << mbOK, 0);  
//    exit(1);  
{  
  
return nbytes;  
{  
-----//  
static int usbopen(void(  
{  
    struct usb_bus      *bus;  
    struct usb_device *dev = 0;  
  
    usb_init();  
    usb_find_busses();  
    usb_busses=usb_get_busses();  
    usb_find_devices();  
    for(bus=usb_busses; bus; bus=bus->next) (  
        for(dev=bus->devices; dev; dev=dev->next) (  
            if(dev->descriptor.idVendor == USBDEV_VENDOR && dev->descriptor.idProduct ==  
USBDEV_PRODUCT(  
                break;  
{  
    if(dev(  
        break;
```

```
if(!dev){  
    MessageDlg("Cannot find USB device." , mtError, TMsgDlgButtons() << mbOK, 0);  
//    exit(1);  
    return 1;  
}  
  
usbhandle = usb_open(dev);  
if(!usbhandle){  
    MessageDlg("Cannot open USB device." , mtError, TMsgDlgButtons() << mbOK, 0);  
//    exit(1);  
    return 1;  
}  
Sleep(100);  
return 0;  
}  
-----//  
unsigned char USB_LCD_Command(unsigned char byte(  
{  
    unsigned char res[4];  
    unsigned char cmd[4];  
    cmd[0]=byte;  
    usbtransmit(1, cmdLCD_Command,cmd, res,sizeof(res);(  
    return res[0];  
}  
-----//  
unsigned char USB_LCD_Data(unsigned char byte(  
{  
    unsigned char res[4];  
    unsigned char cmd[4];  
    cmd[0]=byte;  
    usbtransmit(1, cmdLCD_Data,cmd, res,sizeof(res);(  
    return res[0];  
}  
-----//  
unsigned char USB_LCD_Str(unsigned char *data,unsigned char len(  
}
```

```
usbtransmit(0, cmdLCD_Str, cmd, data, len;(  
return 0;  
{  
-----//  
unsigned char USB_LCD_BL(unsigned char byte(  
{  
    unsigned char res[4];  
    unsigned char cmd[4];  
    cmd[0]=byte;  
    usbtransmit(1, cmdLCD_BL,cmd, res,sizeof(res;(  
    return res[0];  
{  
-----//  
unsigned char USB_SWITCH(unsigned char byte(  
{  
    unsigned char res[4];  
    unsigned char cmd[4];  
    cmd[0]=byte;  
    usbtransmit(1, cmdSWITCH,cmd, res,sizeof(res;(  
    return res[0];  
{  
-----//  
#pragma package(smart_init(  
#pragma resource "*.dfm"  
TForm1 *Form1;  
-----//  
__fastcall TForm1::TForm1(TComponent* Owner(  
:    TForm(Owner(  
{  
    hDLL = LoadLibrary("port.dll;("  
    if (hDLL != NULL(  
}  
    soundin = (LPFNDDLLFUNC1)GetProcAddress(hDLL,"SOUNDIN;("  
    soundout = (LPFNDDLLFUNC2)GetProcAddress(hDLL,"SOUNDOUT;("  
    getrate = (LPFNDDLLFUNC3)GetProcAddress(hDLL,"SOUNDGETRATE;("  
    setrate = (LPFNDDLLFUNC4)GetProcAddress(hDLL,"SOUNDSETRATE;("
```

```
setbytes = (LPFNDDLLFUNC6)GetProcAddress(hDLL,"SOUNDSETBYTES;("
capin  = (LPFNDDLLFUNC7)GetProcAddress(hDLL,"SOUNDCAPIN;("
capout = (LPFNDDLLFUNC8)GetProcAddress(hDLL,"SOUNDCAPOUT;("
{
setrate(11025;(
if(getrate() != 11025(
    exit(1;(
setbytes(1;(
if(getbytes() != 1(
    exit(1;(
set=fopen("setting.txt","rt;("
l+=(getc(set)-48)*10;
l+=(getc(set)-48;(
fclose(set;(
if(!usbopen(
    _usb_open=true;
else
    _usb_open=false;
unsigned char data1[]="Ready to use;".
unsigned char data2[]="Press Start;".
if(_usb_open(
}
    USB_SWITCH(0;(
    USB_LCD_BL(1;(
    USB_LCD_Str(data1,strlen(data1);(
    USB_LCD_Command(0xC0;(
    USB_LCD_Str(data2,strlen(data2);(
{
}
-----//
void __fastcall TForm1::StartClick(TObject *Sender(
}
    unsigned char data[16]="Device 1 is off;".
    if(_usb_open(
}
```



```
USB_LCD_Command(0x80;(
USB_LCD_Str(data,strlen(data);(
{
Timer1->Enabled=true;
{
-----//
void __fastcall TForm1::Timer1Timer(TObject *Sender(
}
static int sw=1;
Timer1->Interval=2300;
on1->Enabled=false;
on2->Enabled=false;
on3->Enabled=false;
on4->Enabled=false;
on5->Enabled=false;
off1->Enabled=false;
off2->Enabled=false;
off3->Enabled=false;
off4->Enabled=false;
off5->Enabled=false;
switch (sw) {
case 1 : on1->Enabled=true;off1->Enabled=true;break;
case 2 : on2->Enabled=true;off2->Enabled=true;break;
case 3 : on3->Enabled=true;off3->Enabled=true;break;
case 4 : on4->Enabled=true;off4->Enabled=true;break;
case 5 : on5->Enabled=true;off5->Enabled=true;break;
}
unsigned char buffer[200000];[
soundin(buffer,2*11025;(
int i,maxx=128,level,start,end,len,maxx1=128,maxx2=128;
float p;
for(i=0;i<2*11025;i(++
if(buffer[i]>maxx(
maxx=buffer[i];[
level=128+(maxx-128)*.25;
for(start=0;start<2*11025;start(++
if(buffer[start]>level(
break;
for(end=2*11025-1;end>0;end(--
if(buffer[end]>level(
```

```
len=end-start;
```

```
int t;
```

```
for(p=0,i=start;i<end;i++)
```

```
}
```

```
t=buffer[i];
```

```
p+=abs((t-128))*abs((t-128);
```

```
{
```

```
p/=len;
```

```
for(i=start;i<start+len*.3;i++)
```

```
if(buffer[i]>maxx1(
```

```
maxx1=buffer[i];
```

```
for(i=start+len*.6;i<end;i++)
```

```
if(buffer[i]>maxx2(
```

```
maxx2=buffer[i];
```

```
l=l/100;
```

```
float f;
```

```
f=maxx1-maxx2;
```

```
f=f/maxx;
```

```
int onoff;
```

```
if((f>1)&&(maxx1>maxx2)&&(p>10((
```

```
onoff=0;
```

```
else if((f<1)&&(p>10((
```

```
onoff=1;
```

```
else
```

```
onoff=2;
```

```
switch (sw) {
```

```
case 1 : if(onoff==1)on1->Checked=true;else if(onoff==0)off1->Checked=true;break;
```

```
case 2 : if(onoff==1)on2->Checked=true;else if(onoff==0)off2->Checked=true;break;
```

```
case 3 : if(onoff==1)on3->Checked=true;else if(onoff==0)off3->Checked=true;break;
```

```
case 4 : if(onoff==1)on4->Checked=true;else if(onoff==0)off4->Checked=true;break;
```

```
case 5 : if(onoff==1)on5->Checked=true;else if(onoff==0)off5->Checked=true;break;
```

```
{
```

```
sw++;
```

```
if(sw==6(
```

```
sw=1;
```

```
Byte switches=0;
```

```
char c1,c2,c3;
```

```
}  
    if(on1->Checked(  
}
```

```
    switches=0;  
    switches |= 0x01;  
    c1='1';  
    c2='n';  
    c3;' '=
```

```
{  
else  
}
```

```
    c1='1';  
    c2='f';  
    c3='f';
```

```
{  
    if(on2->Checked(  
}
```

```
    switches |= 0x02;  
    c1='2';  
    c2='n';  
    c3;' '=
```

```
{  
else  
}
```

```
    c1='2';  
    c2='f';  
    c3='f';
```

```
{  
    if(on3->Checked(  
}
```

```
    switches |= 0x04;  
    c1='3';  
    c2='n';  
    c3;' '=
```

```
{  
else  
}
```

```
    c1='3';  
    c2='f';  
    c3='f';
```

```
}
    switches |= 0x08;
    c1='4;
    c2='n;
    c3;' '=
{
else
}

    c1='4;
    c2='f;
    c3='f;
{
    if(on5->Checked(
}
    switches |= 0x10;
    c1='5;
    c2='n;
    c3;' '=
{
else
}

    c1='5;
    c2='f;
    c3='f;
{

    USB_SWITCH(switches;(
    USB_LCD_Command(0x87;(
    USB_LCD_Data(c1;(
    USB_LCD_Command(0x8D;(
    USB_LCD_Data(c2;(
    USB_LCD_Command(0x8E;(
    USB_LCD_Data(c3;(
{
{
-----//
void __fastcall TForm1::ExitClick(TObject *Sender(
}
```

جهت خرید فایل word به سایت www.kandoo.cn.com مراجعه کنید

یا با شماره های ۰۹۳۶۶۰۲۷۴۱۷ و ۰۹۳۶۶۴۰۶۸۵۷ و ۰۶۶۴۱۲۶۰-۵۱۱ تماس حاصل نمایید

```
{
```

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

```
void __fastcall TForm1::FormCloseQuery(TObject *Sender, bool &CanClose(
```

```
}
```

```
if(_usb_open(
```

```
    USB_LCD_BL(0;(
```

```
{
```

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

www.kandoo.cn.com

www.kandoo.cn.com

www.kandoo.cn.com