#include <C8051F020.h>

#define NOKEY 255

#define uchar unsigned char

void SYSCLK\_Init**(**void**)**

**{**

int i**;**

OSCXCN **=** 0x67**;**

**for(**i**=**0**;** i**<**256**;** **++**i**);**

**while(** **!(**OSCXCN **&** 0x80**)** **);**

OSCICN **=** 0x88**;**

**}**

void PORT\_Init**(**void**)**

**{**

SYSCLK\_Init**();**

EMI0CF **=** 0x1f**;**

XBR2 **=** 0x42**;**

P0MDOUT **=** 0xc0**;**

P1MDOUT **=** 0xff**;**

P2MDOUT **=** 0xff**;**

P3MDOUT **=** 0xff**;**

**}**

void Delay**(**int k**)**

**{**

int i**;**

**for(**i**=**0**;** i**<**k**;** **++**i**);**

**}**

unsigned char xdata seg \_at\_ **(**0x8000**);**

unsigned char xdata cs \_at\_ **(**0x8001**);**

// increase length of segs

const unsigned char code segs**[]** **=**

**{**0xc0**,** 0xf9**,** 0xa4**,** 0xb0**,** 0x99**,** 0x92**,** 0x82**,** 0xf8**,**

0x80**,** 0x90**,** 0x88**,** 0x83**,** 0xc6**,** 0xa1**,** 0x86**,** 0x8e**,** 0xff**};**

const unsigned char code css**[]** **=** **{**0x7**,** 0xb**,** 0xd**,** 0xe**};**

const uchar code dec**[]** **=** **{**0**,** 0**,** 1**,** 0**,** 2**,** 0**,** 0**,** 0**,** 3**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0**};**

const uchar code trans**[]** **=** **{**0xc**,** 9**,** 5**,** 1**,** 0xd**,** 0**,** 6**,** 2**,** 0xe**,**

0xa**,** 7**,** 3**,**0xf**,**0xb**,**8**,** 4**};**

uchar getkey**()** //get key number

**{**

uchar i**;**

uchar key**;**

P4 **=** 0x0f**;**

Delay**(**100**);**

i **=** **~**P4 **&** 0x0f**;**

**if(**i **==** 0**)**

**return** NOKEY**;**

key **=** dec**[**i**]** **\*** 4**;**

Delay**(**1000**);**

P4 **=** 0xf0**;**

Delay**(**100**);**

i **=** **~**P4**;**

i **>>=** 4**;**

**if(**i **==** 0**)**

**return** NOKEY**;**

key **+=** dec**[**i**];**

key **=** trans**[**key**];**

**return** key**;**

**}**

void main**(**void**)**

**{**

unsigned int num **=** 0**;**

unsigned int flag **=** 1**;**

unsigned char i**;**

unsigned char pos**[**4**]** **=** **{**16**,**16**,**16**,**16**};** // pos array

WDTCN **=** 0xde**;**

WDTCN **=** 0xad**;**

PORT\_Init**();**

**while(**1**){**

num **=** getkey**();**

**if(**num **==** NOKEY**)**

flag **=** 1**;**

**if(**num **!=** NOKEY **&&** flag **==** 1**){**

//uchar temp0 = num, temp = num;

//while(temp0 == temp){ // verify input

// temp = getkey();

pos**[**3**]** **=** pos**[**2**];**

pos**[**2**]** **=** pos**[**1**];**

pos**[**1**]** **=** pos**[**0**];**

pos**[**0**]** **=** num**;**

flag **=** 0**;**

**}**

**for(**i**=**0**;** i**<**4**;** **++**i**){**

seg **=** segs**[**pos**[**i**]];**

//else seg = segs[16];

cs **=** css**[**i**];**

num **/=** 16**;**

Delay**(**1000**);**

**}**

**}**

**}**

**之前存在问题的部分代码：**

uchar temp0 **=** num**,** temp **=** num**;**

**while(**temp0 **==** temp**){** // verify input

temp **=** getkey**();**

**}**

#include <C8051F020.h>

#define NOKEY 255

#define uchar unsigned char

unsigned char xdata seg \_at\_ **(**0x8000**);**

unsigned char xdata cs \_at\_ **(**0x8001**);**

int sec**;**

int min**;**

int msec**;**

int count**;**

const unsigned char code segs**[]** **=**

**{**0xC0**,** 0xF9**,** 0xA4**,** 0xB0**,** 0x99**,** 0x92**,** 0x82**,** 0xF8**,**

0x80**,** 0x90**,** 0x88**,** 0x83**,** 0xC6**,** 0xA1**,** 0x86**,** 0x8E**};**

const unsigned char code css**[]** **=** **{**0x7**,** 0xB**,** 0xD**,** 0xE**};**

void SYSCLK\_Init**()**

**{**

int i**;**

OSCXCN**=**0x67**;**

**for(**i**=**0**;**i**<**256**;**i**++);**

**while(!(**OSCXCN**&**0x80**));**

OSCICN**=**0x88**;**

**}**

void PORT\_Init **(**void**)**

**{**

SYSCLK\_Init**();**

EMI0CF **=** 0x1F**;**

XBR2 **=** 0x42**;**

P0MDOUT **=** 0xC0**;**

P1MDOUT **=** 0xFF**;**

P2MDOUT **=** 0xFF**;**

P3MDOUT **=** 0xFF**;**

P74OUT **=** 0x80**;**

**}**

void Delay**(**int k**)**

**{**

int i**;**

**for** **(**i **=** 0**;** i **<** k**;** **++**i**);**

**}**

uchar getkey**()**

**{**

uchar i**;**

uchar key**;**

const uchar code dec**[]** **=** **{**0**,** 0**,** 1**,** 0**,** 2**,** 0**,** 0**,** 0**,** 3**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0**,** 0**};**

const uchar code trans**[]** **=** **{**0xC**,** 9**,** 5**,** 1**,** 0xD**,** 0**,** 6**,** 2**,** 0xE**,** 0xA**,** 7**,** 3**,** 0xF**,** 0xB**,** 8**,** 4**};**

P4 **=** 0x0F**;**

Delay**(**100**);**

i **=** **~**P4 **&** 0x0F**;**

**if** **(**i **==** 0**)** **return** NOKEY**;**

key **=** dec**[**i**]** **\*** 4**;**

Delay**(**1000**);**

P4 **=** 0xF0**;**

Delay**(**100**);**

i **=** **~**P4**;**

i **>>=** 4**;**

**if** **(**i **==** 0**)** **return** NOKEY**;**

key **=** key **+** dec**[**i**];**

key **=** trans**[**key**];**

**return** key**;**

**}**

void TIMER0\_ISR **(**void**)** interrupt 1

**{**

count**++;**

**if** **(**count **>** 72**){**

count **=** 0**;**

msec**++;**

**if** **(**msec**>**99**)**

**{**

sec**++;**

msec**=**0**;**

**if** **(**sec**>**59**){**

sec**=**0**;**

min**=(**min**+**1**)%**60**;**

**}**

**}**

**}**

**}**

//x0:low, x1:high. Show result

void SHOW\_LED**(**int x0**,**int x1**)**

**{**

uchar i**;**

int j**;**

j **=** x0**;**

**for** **(**i**=**0**;** i**<**2**;** **++**i**){**

seg **=** segs**[**j **%** 10**];**

cs **=** css**[**i**];**

j **/=** 10**;**

Delay**(**1000**);**

**}**

j **=** x1**;**

**for** **(**i**=**0**;** i**<**2**;** **++**i**){**

seg **=** segs**[**j **%** 10**];**

cs **=** css**[**i **+** 2**];**

j **/=** 10**;**

Delay**(**1000**);**

**}**

**}**

void main**(**void**)**

**{**

WDTCN **=** 0xde**;**

WDTCN **=** 0xad**;**

PORT\_Init**();**

//timer initialization

EA **=** 1**;**

TMOD **|=** 0x02**;**

TH0 **=** 0x00**;**

TL0 **=** 0x00**;**

TR0 **=** 1**;**

ET0 **=** 1**;**

P7 **=** 0x00**;**

msec **=** 0**;**

sec **=** 0**;**

min **=** 0**;**

count **=** 0**;**

bit mode **=** 0**;** // transfer mode

**while(**1**){**

int ctrl**;**

int x0**,** x1**;**

int show0**,** show1**;**

**if(**mode**==**0**){**

show0 **=** sec**;**

show1 **=** min**;**

**}**

**else{**

show0 **=** msec**;**

show1 **=** sec**;**

**}**

ctrl **=** getkey**();**

// initialization

**if** **(**ctrl **==** 0xA**){**

uchar temp0 **=** 0xA**,** temp **=** 0xA**;**

uchar countn **=** 0**;**

x0 **=** 0**;**

x1 **=** 0**;**

**while(**countn **<** 4**){**

**while(**temp0 **==** temp**){** // verify input

temp **=** getkey**();**

SHOW\_LED**(**x0**,**x1**);**

**}**

temp0 **=** temp**;**

**if** **(**temp **==** NOKEY**)**

**continue;**

**if** **(**countn **==** 0**){**

**if(**temp **<** 0xA**){**

x0 **=** temp**;**

countn**++;**

**}**

**}**

**else** **if(**countn **==** 1**){**

**if(**mode **==** 0**){**

**if(**temp **<** 0x6**){**

x0 **=** x0 **+** 10**\***temp**;**

countn**++;**

**}**

**}**

**else{**

**if(**temp **<** 0xA**){**

x0 **=** x0 **+** 10**\***temp**;**

countn**++;**

**}**

**}**

**}**

**else** **if(**countn **==** 2**){**

**if(**temp **<** 0xA**){**

x1 **=** temp**;**

countn**++;**

**}**

**}**

**else{**

**if(**temp **<** 0x6**){**

x1 **=** x1 **+** temp**\***10**;**

countn**++;**

**}**

**}**

**}**

ctrl**=**0**;**

**if(**mode **==** 0**){**

count **=** 0**;**

msec **=** 0**;**

sec **=** x0**;**

min **=** x1**;**

**}**

**else{**

count **=** 0**;**

msec **=** x0**;**

sec **=** x1**;**

min **=** 0**;**

**}**

**}**

// switch mode

**if** **(**ctrl**==**0xB**){**

uchar temp **=** 0xB**;**

**while(**temp **==** 0xB**){**

SHOW\_LED**(**show0**,**show1**);**

temp **=** getkey**();**

**}**

mode **=** **~**mode**;**

**}**

// clear to zero without waiting

**if** **(**ctrl **==** 0xC**){**

count **=** 0**;**

msec **=** 0**;**

sec **=** 0**;**

min **=** 0**;**

**}**

// clear to zero and wait for recovering

**if** **(**ctrl **==** 0xF**){**

**while(**1**){**

count **=** 0**;**

msec **=** 0**;**

sec **=** 0**;**

min **=** 0**;**

SHOW\_LED**(**0**,**0**);**

ctrl **=** getkey**();**

**if(**ctrl **==** 0xE**)** // recover

**break;**

**}**

**}**

// stop and wait for recover

**if** **(**ctrl **==** 0xD**){**

int record0**,**record1**,**record2**,**record3**;**

record0 **=** count**;**

record1 **=** msec**;**

record2 **=** sec**;**

record3 **=** min**;**

**while(**1**){**

ctrl **=** getkey**();**

SHOW\_LED**(**show0**,**show1**);**

**if(**ctrl **==** 0xE**)** // recover

**break;**

**}**

count **=** record0**;**

msec **=** record1**;**

sec **=** record2**;**

min **=** record3**;**

**}**

SHOW\_LED**(**show0**,**show1**);**

**}**

**}**

上面为分秒/秒毫秒 模式；下面为时分/分秒 模式，部分代码的改进如下

uchar hourflag **=** 0x0**;** // hour's low bit depose

x0 **=** 0**;**

x1 **=** 0**;**

**while(**countn **<** 4**){**

**while(**temp0 **==** temp**){** // verify input

temp **=** getkey**();**

SHOW\_LED**(**x0**,**x1**);**

**}**

temp0 **=** temp**;**

**if** **(**temp **==** NOKEY**)**

**continue;**

**if** **(**countn **==** 0**){**

**if(**temp **<** 0xA**){**

x0 **=** temp**;**

countn**++;**

**}**

**}**

**else** **if(**countn **==** 1**){**

**if(**temp **<** 0x6**){**

x0 **=** x0 **+** 10**\***temp**;**

countn**++;**

**}**

**}**

**else** **if(**countn **==** 2**){**

**if(**temp **<** 0xA**){**

x1 **=** temp**;**

**hourflag = temp;**

countn**++;**

**}**

**}**

**else{**

**if(mode == 1)**

**if(temp < 0x6){**

**x1 = temp;**

**countn++;**

**}**

**else{**

**if(temp < 0x2 && hourflag > 0x3 || temp < 0x3 && hourflag < 0x4){**

**x1 = temp;**

**countn++;**

**}**

**}**