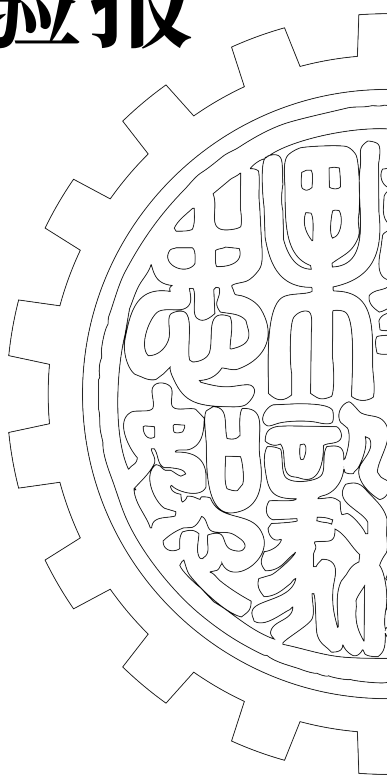
嵌入式系统实验报告

吴思源 自动化钱 71 班

2019 年 11 月 16 日



XI'AN JIAOTONG UNIVERSITY

目录

§1 实验内容

编程调试，实现在瑞萨 RL78G13 实验板 LCD 四行分别显示：姓名、学号、班级、年-月-日，并且实现通过键盘设定年月日；

§2 实验步骤

实现了 LCD 实验板的四行分别显示: 姓名、学号、班级、年-月-日。对于年月日信息，可以实现动态显示，并且实现了输出情况下修改输入数值。

2.1 模块学习

对于瑞萨 RL78G13 实验板的 LCD 模块，可以显示 4 行，每行必须输入 16 个字符。由于每个汉字占两个字符，因此每行不能超过 8 个汉字。对于显示数字和英文字母等 ASCII 码字符，需要将数字和字母转化为 ACSII 码显示。LCD 屏上一共可以显示 64 个字符，0 ~ 15 为第一行，16 ~ 31 为第二行，32 ~ 47 为第三行，48 ~ 63 为最后一行。

显示信息 可以使用 `lcd_display` 函数实现，这个函数一次性在 LCD 板上按照顺序显示所有信息。

设置光标 可以使用 `CursorSet` 函数实现，这个函数可以将光标设置到 LCD 屏上的某一特定位置，之后从这个位置开始输入。

显示单个字符 可以使用`lcd_write` 函数实现，这个函数输出送入的 ASCII 码对应的字符。因此这个函数不可以用作中文显示。

2.2 显示静态信息

在程序中将送入 LCD 显示的信息分成静态信息与动态信息，静态信息在 LCD 板开启时显示一次，动态信息每次发生改变即刷新。当开启系统时即显示静态信息，静态信息显示的部分代码如下

```
// 初始显示静态信息
lcd_display(0,
                    ”姓名␣␣吴思源␣␣␣␣”
                    ”学号：2171310846”
                    ”班级␣␣自动化钱71”
                    ”----年--月--日␣␣”);
                    // 48 49 50 51 54 55 58 59
```

2.3 显示动态信息

为了动态输入年月日信息，需要动态显示信息。这里主要使用了 `CursorSet` 和 `lcd_write` 来实现动态输入显示的功能。

首先将所有数字放入一个字符串中，做成一个数字表，包含 0 ~ 9，方便之后调用[1]，代码如下

```
        unsigned char *number=”0123456789”;
```

声明两个指针变量 `*p1` 刚刚声明的字符串 ”0123456789”，`*p` 指向字符串中的第一个位置，通过改变 `*p` 指向的位置，可以方便地调整使用不同的字符，且必为 ASCII 码。

然后不清空 LCD 目前正在显示的字符，只将 LCD 光标设置到第 48 个字符处[2]。

之后进行键盘扫描，读取键盘此时的输入。一般来说，键盘输入不同类型字符时系统会有不同的响应，键盘没有输入时也会返回某个字符，因此需要对键盘的输入进行分类判断。

[1] 也便于向函数输入 ASCII 码

[2] 第四行的第一个位置

2.3.1 有效数字

当输入为有效的数字时，应该先进行 60ms 的延时，以防止下次也读取到这个输入。然后再光标处写入对应的输入。这里使用指针实现：由于读取到的输入应为一个数字，为 uint 类型，不是 char，因此向 LCD 屏幕上写入的应该是上述字符串对应位置的字符。送入 LCD 显示之后，应该将对应的变量清零，防止抖动。

通过判断，可以使有效数字的输入可以刚好隔过’年’、’月’、’日’三个字符。

```
        delay(600);
        p = p1 + key;
        lcd_write(*p, 1);
        i++;
        num_keyboard = 0;
```

2.3.2 退格符

当输入为退格符号时，将输入退一格显示，并输出 -，实现输错情况下的纠错。详见视频演示。

```
        delay(600);
        i--;
        CursorSet(i);
        lcd_write('-', 1);
        CursorSet(i);
        num_keyboard = 0;
```

2.3.3 清零符

当输入为清零符时，将 LCD 屏上动态的所有输入清零，并且将光标设置到第 48 个字符处，将数字变量清零。清零过程详见视频。

```
        lcd_display(0,
        "姓名␣␣吴思源␣␣␣␣"
        "学号：2171310846"
        "班级␣␣自动化钱71"
        "----年--月--日␣␣");

        //将光标设置到第 48 个字符处
        i = 48;
        CursorSet(i);

        num_keyboard = 0;
```

2.4 调试程序

一个非常容易忘掉的地方,是将键盘的对应端口文件写入r_cg_port.c文件中，使键盘的输入端口有效。

另外一个容易出错的地方，是合理处理看门狗程序。要么去掉看门狗程序，使其不起作用；要么定时喂狗，防止看门狗程序重启系统。

§3 附录：代码

3.1 主函数部分

```
#include "r_cg_macrodriver.h"
#include "r_cg_cgc.h"
#include "r_cg_port.h"
#include "r_cg_wdt.h"
#include "r_cg_lcd.h"
#include "r_cg_userdefine.h"

```

```
/*****************
Global variables and functions
****************/s
unsigned int flag;
unsigned char *string="Xi'an␣␣␣JiaotongUniv.";
unsigned char *number="0123456789";
extern volatile uint8_t num_keyboard;
void lcd_displayXJTU(unsigned char pos,  void *str);
void Keyboard_scan(void);

void delay(int t1)
{
  int x = 0;
  int y = 0;
  for(x = 0; x < t1;x++)
   for(y = 0; y < 1000;y++);
}

/* End user code. Do not edit comment generated here */
void R_MAIN_UserInit(void);
void lcd_Date(void *number);

/***********************************************************************
* Function Name: main
* Description  : This function implements main function.
* Arguments    : None
* Return Value : None
***********************************************************************/
void main(void)
{

        R_MAIN_UserInit();
```

```
40      /* Start user code. Do not edit comment generated here */
41
42          lcd_init( );
43          //LcdFill_Level();
44
45      while (1U)
46      {
47          // 初始显示
48          lcd_display(0,
49                              "姓名␣␣吴思源␣␣␣␣"
50                              "学号：2171310846"
51                              "班级␣␣自动化钱71"
52                              "----年--月--日␣␣");
53                              // 48 49 50 51  54 55  58 59
54          WDTE = 0xAC;
55          delay(1000);
56
57          // 显示日期
58          lcd_Date(number);
59
60          delay(1000);
61          delay(1000);
62
63          NOP();
64          WDTE = 0xAC;
65      }
66
67      /* End user code. Do not edit comment generated here */
68  }
69
70
71  void R_MAIN_UserInit(void)
```

```
{
    /* Start user code. Do not edit comment generated here */
    EI();
    /* End user code. Do not edit comment generated here */
}

/* Start user code for adding.
Do not edit comment generated here */

void lcd_Date(void *number)
{
        unsigned char j, i;
        unsigned char *p, *p1;
        unsigned int key;
        p = p1 = (unsigned char *)number;
        flag = 1;
        i = 48;
        CursorSet(i);
        while (flag)
        {
                WDTE = 0xAC;
                Keyboard_scan();
                key = num_keyboard % 10;
                if (i > 59)
                {
                        CursorSet(48);
                        i = 48;

                }
                else if (i == 52)
                {
                        i = i + 2;
```

```
104                     CursorSet(i);
105
106             }
107             else if (i == 56)
108             {
109                     i = i + 2;
110                     CursorSet(i);
111             }
112             if (num_keyboard < 11 && num_keyboard > 0)
113             {
114                     delay(600);
115                     p = p1 + key;
116                     // *p1 = key;
117                     lcd_write(*p, 1);
118                     i++;
119                     num_keyboard = 0;
120
121             }
122             // 输入退格符
123             else if  (num_keyboard == 13)
124             {
125                     delay(600);
126                     i--;
127                     CursorSet(i);
128                     lcd_write('-', 1);
129                     CursorSet(i);
130                     num_keyboard = 0;
131             }
132             // 输入清零符
133             else if (num_keyboard == 15)
134             {
135                     lcd_display(0,
```

```
136                         "姓名␣␣吴思源␣␣␣␣"
137                         "学号：2171310846"
138                         "班级␣␣自动化钱71"
139                         "----年--月--日␣␣");
140                         i = 48;
141                         CursorSet(i);
142                         num_keyboard = 0;
143                 }
144         }
145
146 }
147
148 /* End user code. Do not edit comment generated here */
```

3.2 键盘模块

```
1
2  void Keyboard_scan(void)
3  {
4
5          KEY_PORT = 0xff;
6          P7.3=0;
7          temp = KEY_PORT;
8          temp=temp&0xf0;          /*check  four  lower  bits*/
9          if(temp!=0xf0)           /*first check*/
10         {
11                 delay(100);              /*delay some time*/
12                 temp = KEY_PORT;
13                 temp=temp&0xf0; /*check  four  lower  bits*/
14                 if(temp!=0xf0)  /*second check*/
15                 {
```

```
                        temp=KEY_PORT;
                        temp=temp&0xf0;
                        switch(temp)
                        {
                        case 0xe0:  num_keyboard=7; break;
                        case 0xd0:  num_keyboard=4; break;
                        case 0xb0:  num_keyboard=1; break;
                        case 0x70:  num_keyboard=10; break;
                        }
                }
                testvalue=num_keyboard;
        }

        KEY_PORT = 0xff;
/*as above*/
        P7.2=0;
        temp=KEY_PORT;
        temp=temp&0xf0;
        if(temp!=0xf0)
        {
                delay(100);
                temp=KEY_PORT;
                temp=temp&0xf0;
                if(temp!=0xf0)
                {  temp=KEY_PORT;
                temp=temp&0xf0;
                switch(temp)
                {
                case 0xe0:  num_keyboard=8; break;
                case 0xd0:  num_keyboard=5; break;
                case 0xb0:  num_keyboard=2; break;
                case 0x70:  num_keyboard=11; break;
```

```
47                  }
48                  }
49          }
50
51          KEY_PORT = 0xff;
    /*as above*/
52          P7.1=0;
53          temp=KEY_PORT;
54          temp=temp&0xf0;
55          if(temp!=0xf0)
56          {
57                  delay(100);
58                  temp=KEY_PORT;
59                  temp=temp&0xf0;
60                  if(temp!=0xf0)
61                  {  temp=KEY_PORT;
62                  temp=temp&0xf0;
63                  switch(temp)
64                  {
65                  case 0xe0:  num_keyboard=9; break;
66                  case 0xd0:  num_keyboard=6; break;
67                  case 0xb0:  num_keyboard=3; break;
68                  case 0x70:  num_keyboard=12; break;
69                  }
70                  }
71          }
72
73          KEY_PORT = 0xff;
    /*as above*/
74          P7.0=0;
75          temp=KEY_PORT;
76          temp=temp&0xf0;
```

```
77          if(temp!=0xf0)
78          {
79                  delay(100);
80                  temp=KEY_PORT;
81                  temp=temp&0xf0;
82                  if(temp!=0xf0)
83                  {  temp=KEY_PORT;
84                  temp=temp&0xf0;
85                  switch(temp)
86                  {
87                  case 0xe0:  num_keyboard=0; break;
88                  case 0xd0:  num_keyboard=13; break;
89                  case 0xb0:  num_keyboard=14; break;
90                  case 0x70:  num_keyboard=15; break;
91                  }
92                  }
93          }
94  }
```