

计算机原理与嵌入式系统设计实验报告（第二次）



# LCD四行分别显示：姓名、学号、班级、年-月-日（通过键盘设定年月日）

## 头文件

#include "r\_cg\_macrodriver.h"

#include "r\_cg\_cgc.h"

#include "r\_cg\_port.h"

#include "r\_cg\_wdt.h"

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

#include "r\_cg\_lcd.h"

#include "r\_cg\_KeyScan.h"

/\* End user code. Do not edit comment generated here \*/

#include "r\_cg\_userdefine.h"

说明：在LCD例程的基础上，增加了key&LED例程当中包括的库函数，在工程文件夹中加入"r\_cg\_KeyScan.h"文件和r\_cg\_KeyScan.c文件，以达到实现键盘输入的目的。

## 全局变量

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

unsigned int i;

extern volatile uint8\_t num\_keyboard;

unsigned char date[16]="\_ 年 月 日 ";

unsigned int x=0;

unsigned char pre=' ';

unsigned int flag=0;

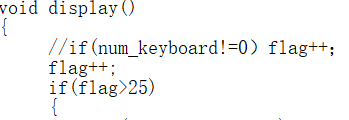
/\* End user code. Do not edit comment generated here \*/

类似于实验一中的全局变量num\_keyboard为键盘输入的值。

Date数组为第四行显示年月日的部分。

X为对年月日输入位置的计数变量

Flag为延时使用的变量，可以通过display（）函数中对其值的变化范围进行限制，可以实现键盘输入的响应时间。

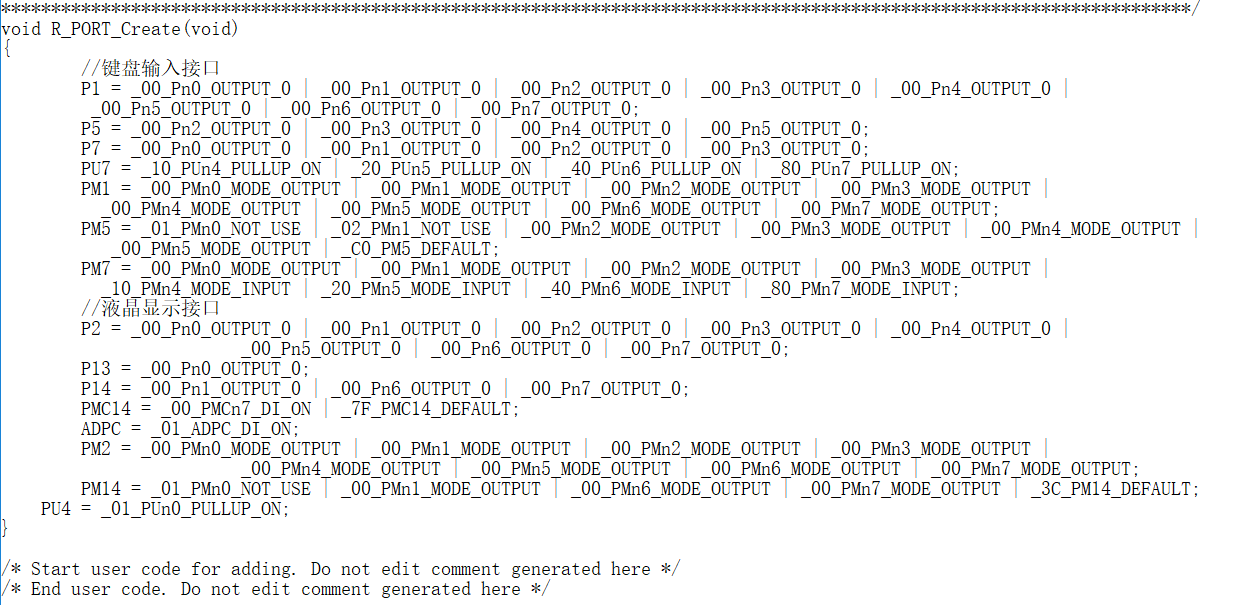


经过实验发现，若flag循环数小于20，则过于灵敏，在按一次按键时可能会输入多个数字；若循环数过大，则键盘的延时会增大，不利于输入。故在此将循环数取为25。

Pre初始化为一个空格字符，在运行时做中间变量存储显示的值。

i为源例程程序当中的变量，为主程序中原有的一个计数变量，在这里不做使用，特此说明。

## 端口定义



将key&LED例程当中的端口定义粘贴过来即可。

## 子函数

void R\_MAIN\_UserInit(void);

void display();

void getnext();

void getpre();

void R\_MAIN\_UserInit(void);函数是原始定义的初始化函数

void display();是进行输入、显示的主要函数

void getnext();和void getpre();分别是获取下一个该输入的位置和获取上一个输入的位置（要避开已有的“年 月 日”）

## 主程序

void main(void)

{

hdwinit();

R\_MAIN\_UserInit();

lcd\_init();

num\_keyboard=0;

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

//lcd\_init( );

//LcdFill\_Level();

while (1U)

{

//for(i = 0;i < 2000; i++)

lcd\_display(0," 任泽华 "

" 自动化71班 "

" 2171411498 ");

lcd\_display(48,date);

Keyboard\_scan();

if(num\_keyboard!=0)

display();

NOP();

WDTE = 0xAC;

}

/\* End user code. Do not edit comment generated here \*/

}

程序说明：

1. 在LCD例程的基础上，增加了key&LED例程中的键盘模块。
2. 在while循环中，首先播放固定的内容，即姓名学号；再播放键盘输入模块（年月日）。
3. 增加键盘输入，改变（年月日）模块的参数，实现键盘输入。

## Display（）函数

void display()

{

//if(num\_keyboard!=0）flag++；

flag++;

if(flag>25)

{

if(num\_keyboard>10)

{

switch(num\_keyboard)

{

case(11):

{

getpre();

date[x]=' ';

break;

}

case(13):

{

date[x]=pre;

getpre();

pre=date[x];

date[x]=95;

break;

}

}

}

else

{

date[x]=num\_keyboard+47;

x++;

getnext();

if(x!=0) date[x]=95;//显示\_指示下一位输入位置

}

flag=0;

num\_keyboard=0;

}

}

这是键盘输入显示的主要部分，首先通过flag实现延时，避免过于灵敏。通过键盘输入的参数进行选择，定义键盘的顺序1~10分别对应字符0~9。在输入后下一位显示“\_”指示当前的输入位。另外定义了删除键和选择位置键，它们对应的按键分别是11、13。

## Getnext（）和getpre（）函数

void getnext()

{

switch(x)

{

case(4):x=x+2;break;

case(8):x=x+2;break;

case(12):x=0;break;

default:break;

}

}

void getpre()

{

switch(x)

{

case(6):x=x-3;break;

case(10):x=x-3;break;

case(0):x=11;break;

default:x--;

}

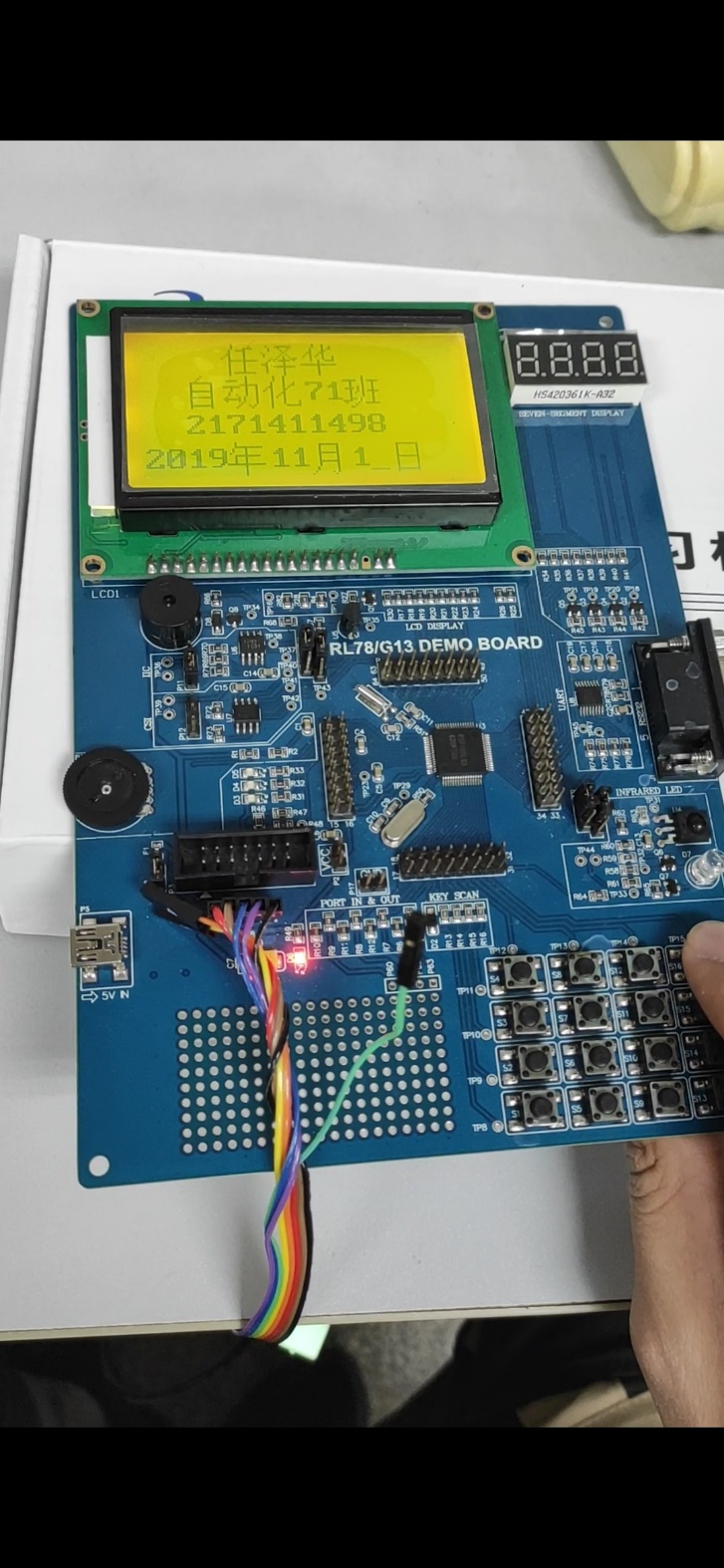
}

获取下一个该输入的位置和获取上一个输入的位置

（避开已有的“年 月 日”）

## 实验照片：

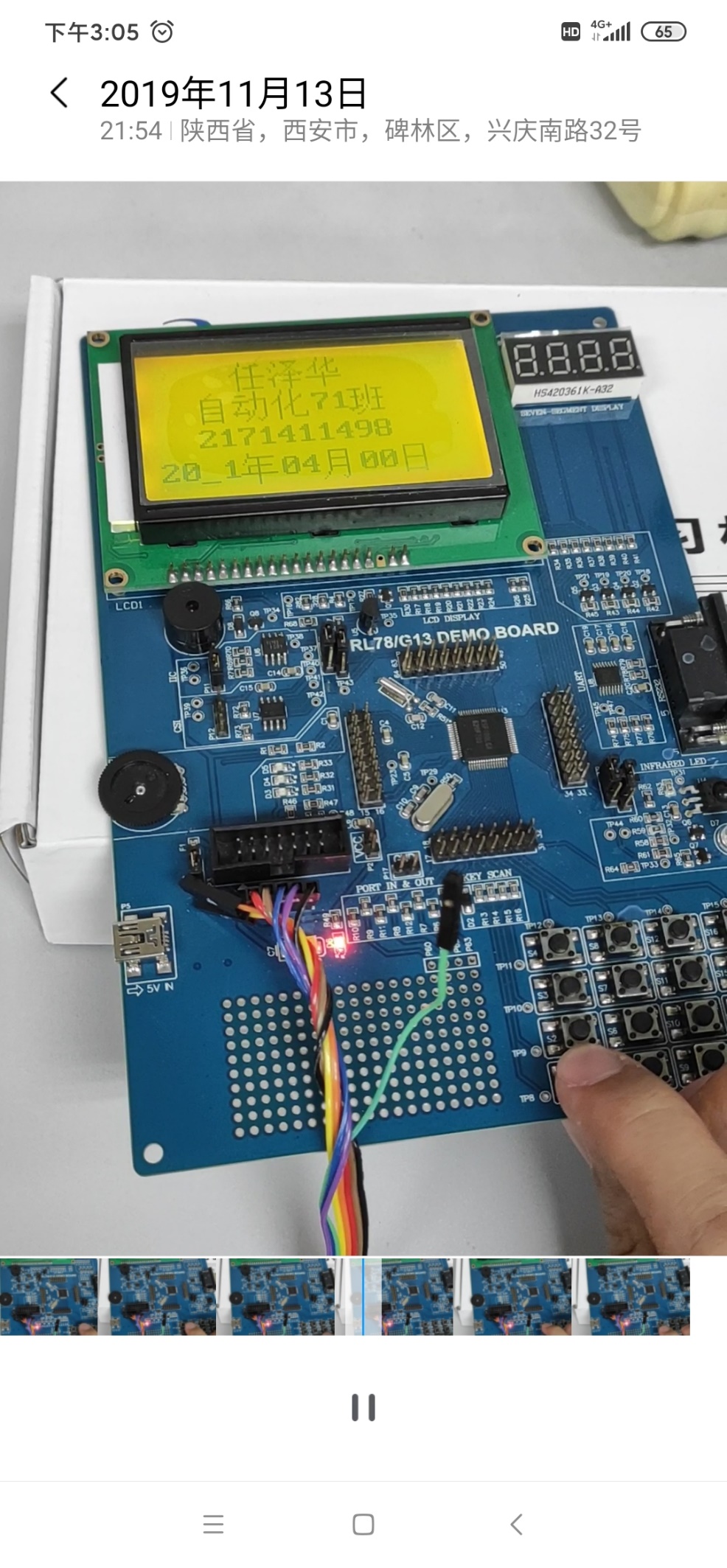
输入日期：



覆盖日期：



改变输入位置（左移）：



删除输入：

