



《单片机实践》实验报告书

第二次实验：矩阵按键操作实验

姓 名 林宇航

学 号 201906060308

指导老师 郭方洪

专业班级 自动化1901

实验地点与日期 信 B116 2021.11.10

**一、实验目的**

1.熟悉和掌握矩阵式键盘的工作原理、电路设计和软件编程方法；

2.熟悉和掌握矩阵式键盘的行列扫描法；

3.掌握键盘延时消抖的软件办法。

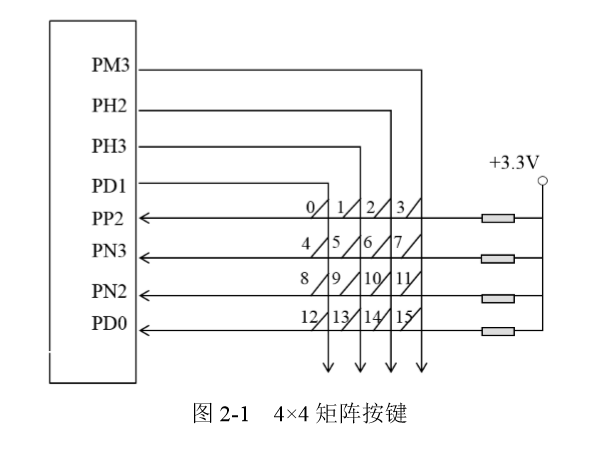
**二、实验内容**

1.共16个按键，实现按下按键后，对应LED灯点亮。

2.实现同时按下两个按键时的检测和响应。

**三、实验步骤**

矩阵按键又称为行列式按键，他是用4条I/O线作为行线，4条I/O线作为列线组成的 按键，在行线和列线的每一个交叉点上，设置一个按键。矩阵按键原理如下图所示：



这样矩阵中就有 4×4 个按键，这种行列式键盘结构能够有效地提高单片机系统中 I/O 口的利用率。

1、按键识别原理

在某一时刻只让一条列线处于低电平，其余列线均处于高电平，则当这一列有键按下时，该键所在的行电平将会由高电平变为低电平，可判定该列相应的行有键按下。当第0列处于低电平时，逐行查找是否有行线变低，若有，则第0列与该行的交叉点按键按下；若无，则表示第0列无键按下，再让下一列处在低电平，依此循环，这种方式称为键盘扫描。

2、软件消抖原理

如果按键较多，常用软件方法去抖，即检测出键闭合后执行一个延时程序，5ms～20ms的延时，让前沿抖动消失后再一次检测键的状态，如果仍保持闭合状态电平，则确认为真正有键按下。当检测到按键释放后，也要给5ms～20ms的延时，待后沿抖动消失后才能转入该键的处理程序。一般来说，软件消抖的方法是不断检测按键值，直到按键值稳定。实现方法：假设未按键时输入1，按键后输入为0，抖动时不定。可以做以下检测：检测到按键输入为0之后，延时5ms～20ms，再次检测，如果按键还为0，那么就认为有按键输入。延时的5ms～20ms恰好避开了抖动期。

实验程序流程图如下 ：

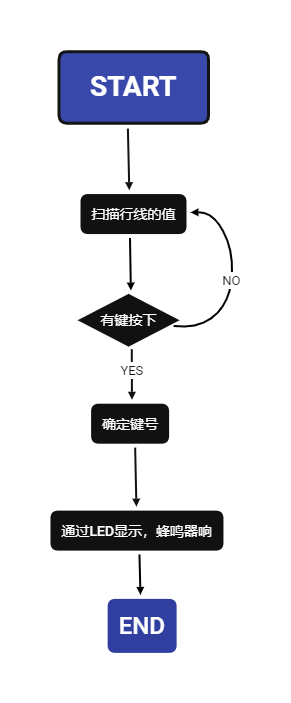


图2- 1 流程图

**四、实验结果**

实现了按下按键后蜂鸣器发出声音并点亮相应编码的指示灯。

**五、实验思考与总结**

通过本次实验，我对单片机按键检测的理解进一步提升，在平时的嵌入式开发中，编写菜单的时候，为了避免按下一次按键多次检测到，我一般会加入一个while语句，如果按键一直按下的话就会陷入局部的循环，直到按键弹起。而如今我明白了这种方法的局限性，检测多个按键按下的时候就显得很鸡肋。实验中的方法是延时一段时间，这样检测多个按键按下就十分方便，而且还可以检测长按。

**六、实验代码、注释及现象**

**基础部分代码：**

**1.main.c**

**#include** "headfile.h"

/\*\*

\* main.c

\*/

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

{

matKey\_gpio\_init();

beep\_gpio\_init();

led\_gpio\_init();

uint8\_t tempdata = 0;

uint8\_t led\_flag=0;

GPIO\_L(List1);

while(1)

{

GPIO\_L(List1);

scan\_key(&tempdata);

if(tempdata)

{

led\_flag = 1+4\*(tempdata - 1);

LED\_switch\_on(led\_flag);

Delay(200);

}

GPIO\_H(List1);

GPIO\_L(List2);

scan\_key(&tempdata);

if(tempdata)

{

led\_flag = 2+4\*(tempdata - 1);

LED\_switch\_on(led\_flag);

Delay(200);

}

GPIO\_H(List2);

GPIO\_L(List3);

scan\_key(&tempdata);

if(tempdata)

{

led\_flag = 3+4\*(tempdata - 1);

LED\_switch\_on(led\_flag);

Delay(200);

}

GPIO\_H(List3);

GPIO\_L(List4);

scan\_key(&tempdata);

if(tempdata)

{

led\_flag = 4+4\*(tempdata - 1);

LED\_switch\_on(led\_flag);

Delay(200);

}

GPIO\_H(List4);

}

return 0;}

**2.headfile.h**

/\*

\* headfile.h

\*

\* Created on: 2021年11月10日

\* Author: Fly

\*/

**#ifndef** USER\_INC\_HEADFILE\_H\_

**#define** USER\_INC\_HEADFILE\_H\_

**#include** <stdint.h>

**#include** <stdbool.h>

**#include** "inc/hw\_ints.h"

**#include** "inc/hw\_memmap.h"

**#include** "inc/hw\_nvic.h"

**#include** "inc/hw\_types.h"

**#include** "driverlib/gpio.h"

**#include** "driverlib/pwm.h"

**#include** "driverlib/interrupt.h"

**#include** "driverlib/pin\_map.h"

**#include** "driverlib/systick.h"

**#include** "driverlib/sysctl.h"

**#include** "driverlib/timer.h"

**#include** "utils/uartstdio.h"

//////////////////////////////////////USER

**#include** "Fly\_gpio.h"

**#include** "Fly\_pwm.h"

**#include** "Fly\_pit.h"

**#endif** /\* USER\_INC\_HEADFILE\_H\_ \*/

**3.gpio.c**

/\*

\* gpio.c

\*

\* Created on: 2021年11月10日

\* Author: Fly

\*/

**#include** "Fly\_gpio.h"

**void** **beep\_gpio\_init**(**void**)

{

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOM); //使能gpio时钟 输出

**GPIODirModeSet**(GPIO\_PORTM\_BASE,GPIO\_PIN\_5,GPIO\_DIR\_MODE\_OUT); //列1

**GPIOPadConfigSet**(GPIO\_PORTM\_BASE,GPIO\_PIN\_5,GPIO\_STRENGTH\_8MA\_SC,GPIO\_PIN\_TYPE\_STD);

**GPIOPinWrite**(GPIO\_PORTM\_BASE,GPIO\_PIN\_5,0x00);

}

**void** **matKey\_gpio\_init**(**void**) //矩阵按键初始化

{

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOD); //使能gpio时钟 输出

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOH);

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOM);

**GPIODirModeSet**(GPIO\_PORTD\_BASE,GPIO\_PIN\_1,GPIO\_DIR\_MODE\_OUT); //列1

**GPIODirModeSet**(GPIO\_PORTH\_BASE,GPIO\_PIN\_3,GPIO\_DIR\_MODE\_OUT); //列2

**GPIODirModeSet**(GPIO\_PORTH\_BASE,GPIO\_PIN\_2,GPIO\_DIR\_MODE\_OUT); //列3

**GPIODirModeSet**(GPIO\_PORTM\_BASE,GPIO\_PIN\_3,GPIO\_DIR\_MODE\_OUT); //列4

**GPIOPadConfigSet**(GPIO\_PORTD\_BASE,GPIO\_PIN\_1,GPIO\_STRENGTH\_8MA\_SC,GPIO\_PIN\_TYPE\_STD);

**GPIOPadConfigSet**(GPIO\_PORTH\_BASE,GPIO\_PIN\_3,GPIO\_STRENGTH\_8MA\_SC,GPIO\_PIN\_TYPE\_STD);

**GPIOPadConfigSet**(GPIO\_PORTH\_BASE,GPIO\_PIN\_2,GPIO\_STRENGTH\_8MA\_SC,GPIO\_PIN\_TYPE\_STD);

**GPIOPadConfigSet**(GPIO\_PORTM\_BASE,GPIO\_PIN\_3,GPIO\_STRENGTH\_8MA\_SC,GPIO\_PIN\_TYPE\_STD);

**GPIOPinWrite**(GPIO\_PORTD\_BASE,GPIO\_PIN\_1,0xff);

**GPIOPinWrite**(GPIO\_PORTH\_BASE,GPIO\_PIN\_3,0xff);

**GPIOPinWrite**(GPIO\_PORTH\_BASE,GPIO\_PIN\_2,0xff);

**GPIOPinWrite**(GPIO\_PORTM\_BASE,GPIO\_PIN\_3,0xff);

////////////////////////////////////////////////////////////////////////////////////////

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPIOP); //使能gpio时钟 输入

**SysCtlPeripheralEnable**(SYSCTL\_PERIPH\_GPION);

**GPIODirModeSet**(GPIO\_PORTP\_BASE,GPIO\_PIN\_2,GPIO\_DIR\_MODE\_IN); //行1

**GPIODirModeSet**(GPIO\_PORTN\_BASE,GPIO\_PIN\_3,GPIO\_DIR\_MODE\_IN); //行2

**GPIODirModeSet**(GPIO\_PORTN\_BASE,GPIO\_PIN\_2,GPIO\_DIR\_MODE\_IN); //行3

**GPIODirModeSet**(GPIO\_PORTD\_BASE,GPIO\_PIN\_0,GPIO\_DIR\_MODE\_IN); //行4

**GPIOPadConfigSet**(GPIO\_PORTP\_BASE,GPIO\_PIN\_2,GPIO\_STRENGTH\_2MA,GPIO\_PIN\_TYPE\_STD\_WPU);

**GPIOPadConfigSet**(GPIO\_PORTN\_BASE,GPIO\_PIN\_3,GPIO\_STRENGTH\_2MA,GPIO\_PIN\_TYPE\_STD\_WPU);

**GPIOPadConfigSet**(GPIO\_PORTN\_BASE,GPIO\_PIN\_2,GPIO\_STRENGTH\_2MA,GPIO\_PIN\_TYPE\_STD\_WPU);

**GPIOPadConfigSet**(GPIO\_PORTD\_BASE,GPIO\_PIN\_0,GPIO\_STRENGTH\_2MA,GPIO\_PIN\_TYPE\_STD\_WPU);

// GPIOPinWrite(GPIO\_PORTD\_BASE,GPIO\_PIN\_1,0xff);

// GPIOPinWrite(GPIO\_PORTH\_BASE,GPIO\_PIN\_3,0xff);

// GPIOPinWrite(GPIO\_PORTH\_BASE,GPIO\_PIN\_2,0xff);

// GPIOPinWrite(GPIO\_PORTM\_BASE,GPIO\_PIN\_3,0xff);

}**void** **scan\_key**(uint8\_t \* data)

{

**if**(!GPIO\_read(Line1))

{

**while**(!GPIO\_read(Line1))

{

LED8\_on;

beep\_on;

}

beep\_off;

LED8\_off;

\*data = 1;

}

**else** **if**(!GPIO\_read(Line2))

{

**while**(!GPIO\_read(Line2))

{

LED8\_on;

beep\_on;

}

beep\_off;

LED8\_off;

\*data = 2;

}

**else** **if**(!GPIO\_read(Line3))

{

**while**(!GPIO\_read(Line3))

{

LED8\_on;

beep\_on;

}

beep\_off;

LED8\_off;

\*data = 3;

}

**else** **if**(!GPIO\_read(Line4))

{

**while**(!GPIO\_read(Line4))

{

LED8\_on;

beep\_on;

}

beep\_off;

LED8\_off;

\*data = 4;

}

**else**

{

\*data = 0;

}

}**4.gpio.h**

/\*

\* gpio.h

\*

\* Created on: 2021年11月10日

\* Author: Fly

\*/

**#ifndef** USER\_INC\_FLY\_GPIO\_H\_

**#define** USER\_INC\_FLY\_GPIO\_H\_

**#include** "headfile.h"

**#define** LED1 GPIO\_PORTF\_BASE,GPIO\_PIN\_1

**#define** LED2 GPIO\_PORTF\_BASE,GPIO\_PIN\_2

**#define** LED3 GPIO\_PORTF\_BASE,GPIO\_PIN\_3

**#define** LED4 GPIO\_PORTL\_BASE,GPIO\_PIN\_0

**#define** LED5 GPIO\_PORTL\_BASE,GPIO\_PIN\_1

**#define** LED6 GPIO\_PORTL\_BASE,GPIO\_PIN\_2

**#define** LED7 GPIO\_PORTL\_BASE,GPIO\_PIN\_3

**#define** LED8 GPIO\_PORTL\_BASE,GPIO\_PIN\_4

**#define** Line1 GPIO\_PORTP\_BASE,GPIO\_PIN\_2

**#define** Line2 GPIO\_PORTN\_BASE,GPIO\_PIN\_3

**#define** Line3 GPIO\_PORTN\_BASE,GPIO\_PIN\_2

**#define** Line4 GPIO\_PORTD\_BASE,GPIO\_PIN\_0

**#define** List1 GPIO\_PORTD\_BASE,GPIO\_PIN\_1

**#define** List2 GPIO\_PORTH\_BASE,GPIO\_PIN\_3

**#define** List3 GPIO\_PORTH\_BASE,GPIO\_PIN\_2

**#define** List4 GPIO\_PORTM\_BASE,GPIO\_PIN\_3

**#define** GPIO\_H(x) GPIOPinWrite(x,0xff)

**#define** GPIO\_L(x) GPIOPinWrite(x,0x00)

**#define** GPIO\_read(x) GPIOPinRead(x)

**#define** beep GPIO\_PORTM\_BASE,GPIO\_PIN\_5

**#define** beep\_on GPIO\_H(beep)

**#define** beep\_off GPIO\_L(beep)

**#define** LED1\_on GPIO\_H(LED1)

**#define** LED2\_on GPIO\_H(LED2)

**#define** LED3\_on GPIO\_H(LED3)

**#define** LED4\_on GPIO\_H(LED4)

**#define** LED5\_on GPIO\_H(LED5)

**#define** LED6\_on GPIO\_H(LED6)

**#define** LED7\_on GPIO\_H(LED7)

**#define** LED8\_on GPIO\_H(LED8)

**#define** LED1\_off GPIO\_L(LED1)

**#define** LED2\_off GPIO\_L(LED2)

**#define** LED3\_off GPIO\_L(LED3)

**#define** LED4\_off GPIO\_L(LED4)

**#define** LED5\_off GPIO\_L(LED5)

**#define** LED6\_off GPIO\_L(LED6)

**#define** LED7\_off GPIO\_L(LED7)

**#define** LED8\_off GPIO\_L(LED8)

**#define** all\_LED\_off LED1\_off;LED2\_off;LED3\_off;LED4\_off;LED5\_off;LED6\_off;LED7\_off;LED8\_off;

**#define** Delay(x) SysCtlDelay(x\*10000) //

// value循环左移bits位

**#define** rol(value, bits) ((value << bits) | (value >> (**sizeof**(value)\*8 - bits)))

// value循环右移bits位

**#define** ror(value, bits) ((value >> bits) | (value << (**sizeof**(value)\*8 - bits)))

**void** **matKey\_gpio\_init**(**void**);

**void** **beep\_gpio\_init**(**void**);

**void** **scan\_key**(uint8\_t \* data);**#endif** /\* USER\_INC\_FLY\_GPIO\_H\_ \*/