电路原理图，我们可以通过以下步骤设计按键程序，实现4个按键按下后蜂鸣器响：

1. 配置IO口：根据电路原理图，将PE0~PE3和PC8口配置为输入口，将蜂鸣器接口PC3配置为输出口。
2. 设置按键检测条件：由于按键是机械式按键，会出现按下和弹起两个状态，为了避免按键抖动的影响，需要设置按键检测条件，比如连续检测几次按键是否处于按下状态。
3. 检测按键状态：通过读取PE0~PE3口的电平状态，判断按键是否处于按下状态。
4. 触发蜂鸣器：如果四个按键均被按下，将PC3口输出高电平，触发蜂鸣器响起。

针对图2，PNP三极管是低电平导通，当PC8口输出低电平时，PNP三极管导通，蜂鸣器得以通电响起。

下面是基于STM32的示例代码（使用标准库）：

#include "stm32f10x.h"

void GPIO\_Configuration(void)

{

GPIO\_InitTypeDef GPIO\_InitStructure;

// 使能GPIO口时钟 RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOC|RCC\_APB2Periph\_GPIOE, ENABLE);

// 配置PE0~PE3口为输入口

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_0 | GPIO\_Pin\_1 | GPIO\_Pin\_2 | GPIO\_Pin\_3;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_IPU;

GPIO\_Init(GPIOE, &GPIO\_InitStructure);

// 配置PC8口为输入口

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_8;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_IPU;

GPIO\_Init(GPIOC, &GPIO\_InitStructure);

// 配置PC3口为输出口

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_3;

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz;

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_PP;

GPIO\_Init(GPIOC, &GPIO\_InitStructure);

}

void Delay(uint32\_t nCount)

{

for(; nCount != 0; nCount--);

}

int main(void)

{

GPIO\_Configuration();

while (1)

{

uint32\_t key\_state = GPIO\_ReadInputData(GPIOE) & 0x000F;

if(key\_state == 0x000F)

{

GPIO\_SetBits(GPIOC, GPIO\_Pin\_3);

Delay(1000000);

GPIO\_ResetBits(GPIOC, GPIO\_Pin\_3);

Delay(1000000);

}

}

}

以上代码中，**GPIO\_Configuration()**函数用于配置IO口，**Delay()**函数用于延时，**main()**函数中，首先读取PE0~PE3口的电平状态，将其与0x000F做与运算，判断四个按键是否均被按下，如果是，则触发蜂鸣器响起

```

#include "stm32f10x.h" // 包含STM32F10x系列芯片的头文件

void GPIO\_Configuration(void) // 定义GPIO配置函数

{

GPIO\_InitTypeDef GPIO\_InitStructure; // 定义GPIO初始化结构体

// 使能GPIO口时钟

RCC\_APB2PeriphClockCmd(RCC\_APB2Periph\_GPIOC|RCC\_APB2Periph\_GPIOE, ENABLE);

// 配置PE0~PE3口为输入口

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_0 | GPIO\_Pin\_1 | GPIO\_Pin\_2 | GPIO\_Pin\_3; // 设置要初始化的GPIO引脚

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_IPU; // 设置GPIO模式为输入模式，带上拉电阻

GPIO\_Init(GPIOE, &GPIO\_InitStructure); // 初始化GPIOE

// 配置PC8口为输入口

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_8; // 设置要初始化的GPIO引脚

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_IPU; // 设置GPIO模式为输入模式，带上拉电阻

GPIO\_Init(GPIOC, &GPIO\_InitStructure); // 初始化GPIOC

// 配置PC3口为输出口

GPIO\_InitStructure.GPIO\_Pin = GPIO\_Pin\_3; // 设置要初始化的GPIO引脚

GPIO\_InitStructure.GPIO\_Speed = GPIO\_Speed\_50MHz; // 设置GPIO速度为50MHz

GPIO\_InitStructure.GPIO\_Mode = GPIO\_Mode\_Out\_PP; // 设置GPIO模式为推挽输出

GPIO\_Init(GPIOC, &GPIO\_InitStructure); // 初始化GPIOC

}

void Delay(uint32\_t nCount) // 定义延时函数

{

for(; nCount != 0; nCount--); // 空循环，实现延时

}

int main(void)

{

GPIO\_Configuration(); // 调用GPIO配置函数

while (1)

{

uint32\_t key\_state = GPIO\_ReadInputData(GPIOE) & 0x000F; // 读取GPIOE输入引脚的状态，并且只取低4位

if(key\_state == 0x000F) // 判断是否按下按键，如果按下按键，key\_state为0x000F

{

GPIO\_SetBits(GPIOC, GPIO\_Pin\_3); // 设置GPIOC的第3个引脚为高电平，即点亮LED灯

Delay(1000000); // 延时一段时间

GPIO\_ResetBits(GPIOC, GPIO\_Pin\_3);// 复位GPIOC的第3个引脚为低电平，即关闭LED灯

Delay(1000000); // 延时一段时间

}

}

}

```