

目录



1

按键实验硬件连接

2

GPIO输入操作说明

3

按键实验

✓ 按键实验硬件连接

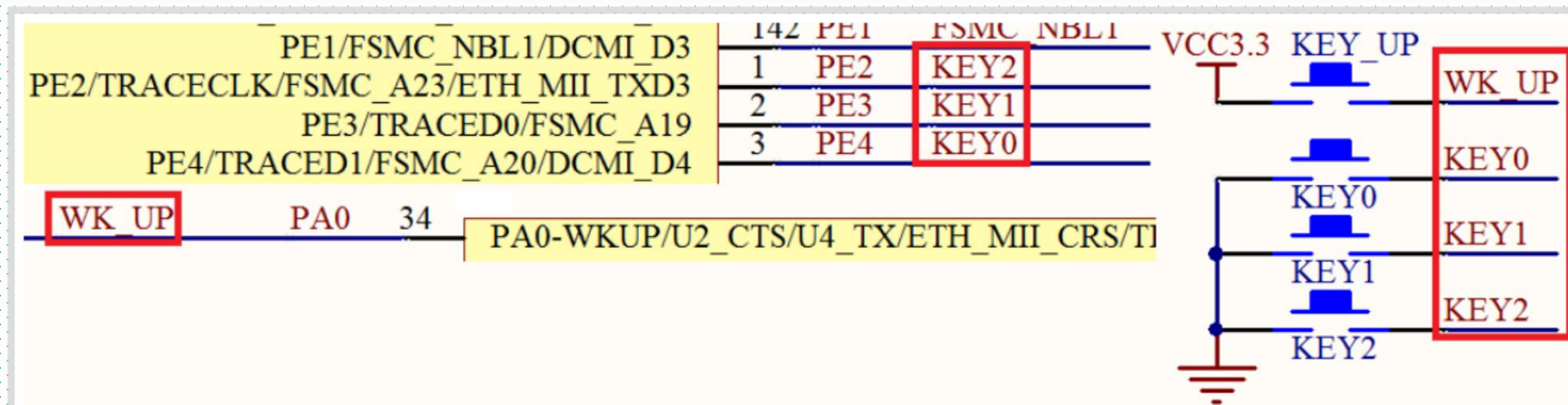


图 8.2.1 按键与 STM32F4 连接原理图

KEY0->PE4 上拉输入
KEY1->PE3 上拉输入
KEY2->PE2 上拉输入
WK_UP->PA0 下拉输入

GPIO输入库函数说明



◆ 读取IO口输入电平调用库函数为:

```
uint8_t GPIO_ReadInputDataBit(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin);
```

◆ 读取IO口输入电平操作寄存器为:

GPIOx_IDR:端口输入寄存器

◆ 使用位带操作读取IO口输入电平方法:

PEin(4) -读取GPIOE.4口电平

PEin(n) -读取GPIOE.n口电平

✓ 3. 按键输入实验



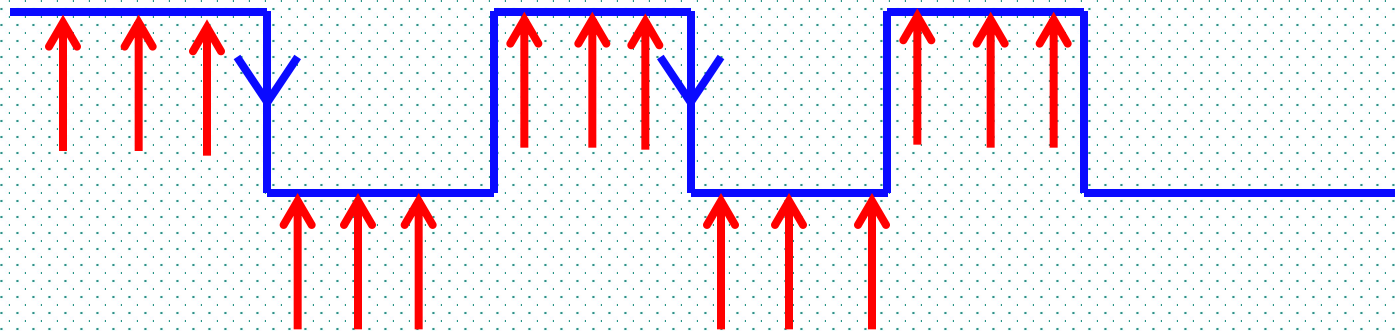
◆ 手把手写按键输入实验。

- 使能按键对应IO口时钟。调用函数：
`RCC_AHB1PeriphClockCmd ();`
- 初始化IO模式：上拉/下拉输入。调用函数：
`GPIO_Init();`
- 扫描IO口电平（库函数/寄存器/位操作）。

✓ 3. 按键输入实验



◆ 按键扫描思路



✓ 3. 按键输入实验



◆每次调用**getValue**函数之后，返回值是多少？

```
int getValue(void)
{
    int flag=0;
    flag++;
    return flag;
}
```

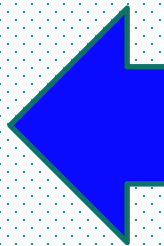
```
int getValue(void)
{
    static int flag=0;
    flag++;
    return flag;
}
```

✓ 3. 按键输入实验



◆ 按键扫描（支持连续按）的一般思路

```
u8 KEY_Scan(void)
{
    if(KEY按下)
    {
        delay_ms(10); //延时10-20ms, 防抖。
        if(KEY确实按下)
        {
            return KEY_Value;
        }
        return 无效值;
    }
}
```



如果我要实现：按键按下，没有松开，只能算按下一次，这个函数无法实现。

✓ 3. 按键输入实验



◆ 按键扫描（不支持连续按）的一般思路

```
u8 KEY_Scan(void)
{
    static u8 key_up=1;
    if (key_up && KEY按下)
    {
        delay_ms(10); //延时，防抖
        key_up=0; //标记这次key已经按下
        if(KEY确实按下)
        {
            return KEY_VALUE;
        }
    }
    }else if(KEY没有按下) key_up=1;
}
```

不支持连续按：就是说，按键按下了，没有松开，只能算一次。

✓ 3. 按键输入实验



◆ 按键扫描（两种模式合二为一）的一般思路

```
u8 KEY_Scan(u8 mode)
{
    static u8 key_up=1;
    if(mode==1) key_up=1;//支持连续按
    if (key_up && KEY按下)
    {
        delay_ms(10);//延时，防抖
        key_up=0;//标记这次key已经按下
        if(KEY确实按下)
        {
            return KEY_VALUE;
        }
    }else if(KEY没有按下) key_up=1;
    return 没有按下
}
```