现代电子系统设计

Psoc 综合实验

超声波测距

学　　号　　2017011589

姓　　名　　吾尔开希

专　　业　　自动化

日　　期　　2019.7.1---7.12

目录

1. 实验内容

利用超声波模块测量距离，将距离信息显示在 EINK 墨水屏上，同时通过串口发送给电脑端。当距离过近时（小于 20cm），触发警报，亮红灯。

2. 设计方案

2.1 超声波测距

超声波模块的使用需要控制好 Trig 引脚，同时准确测量 Echo 信号的长度。用状态机实现对 Trig 的控制和对 Echo 的测量。

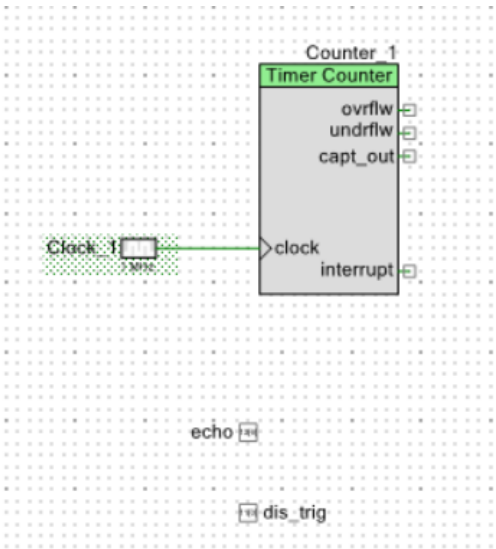
2.2 EINK 显示

得到距离测量值后，调用 EINK 显示函数，详情见流程说明。

3. 电路图

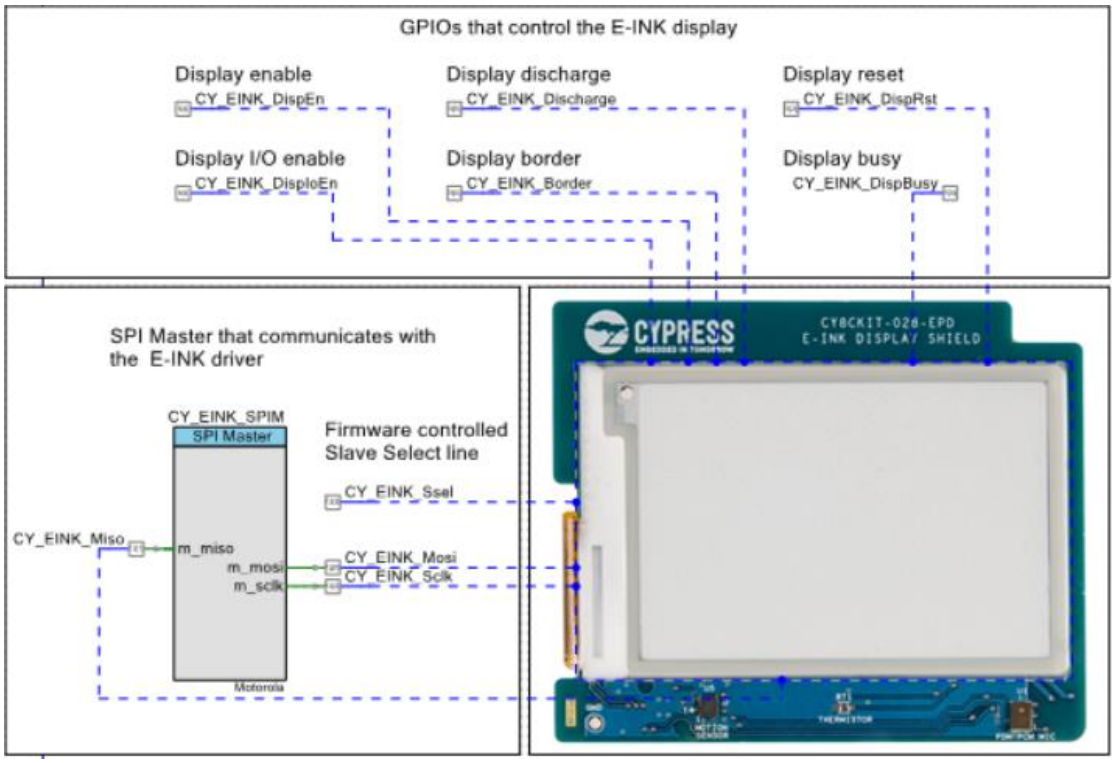
3.1 超声波模块

超声波测距需要控制 dis_trig 输出引脚、echo 输入引脚与一个计数器。计数器用来测量 echo 信号高电平时长。

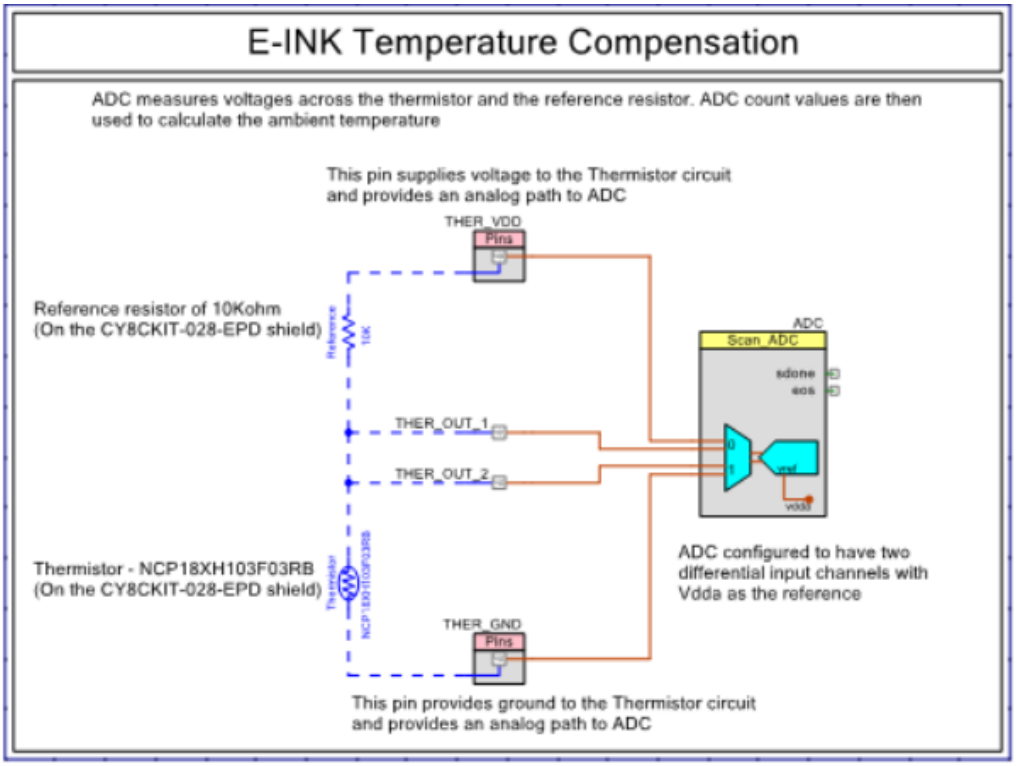


3.2 EINK 显示

EINK 显示的硬件驱动一方面需要一系列的 GPIO 控制，另一方面需要 SPI_MASTER 模块来给它传输信息。



此外，EINK 正常显示还需要环境温度信息。



3.3 LED 灯

LED 灯用一个输出引脚控制即可。

4. 模块选择与参数设置

4.1 超声波模块

超声波测距需要控制 dis_trig 输出引脚、echo 输入引脚与一个计数器。

dis_trig 驱动模式为 strong drive；超声模块的 echo 输出驱动能力较弱，所以 echo 引脚的驱动模式为 resistive pull up。

超声测距模块分辨率为 1mm，对应的 Echo 信号高电平时间：$T_h = 2 \times \frac{0.001}{340} \approx 5.88 \times 10^{-6} s$因此驱动计数器的时钟频率 f>170068Hz，取 f=1MHz。

距离的计算公式$\mathrm{x} = \mathrm{T_h} \times \frac{340}{2} m = 0.017n\ cm$，n 指的是 Echo 信号高电平时间

内计数器的计数次数。

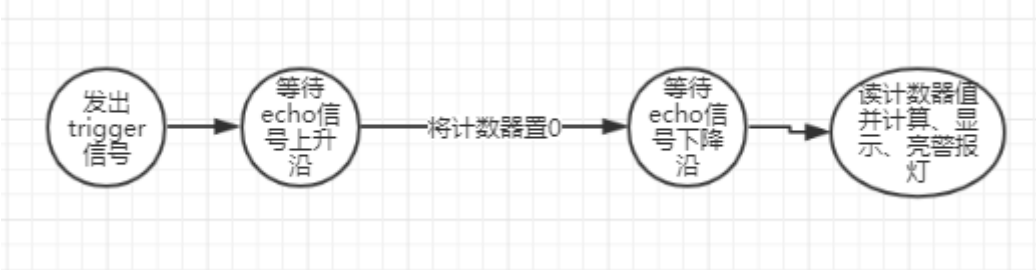
超声测距模块测量最大值 550cm 对应 n=32352，最小值 2cm 对应 n=118，因此计数器的寄存器位数足够（32 位，最大值 65535）

4.2 EINK 模块

EINK 模块的硬件设置已在电路图中说明，没有设置特殊参数。

5. 流程图

5.1 主逻辑流程



主逻辑流程如图所示，在代码中用状态机的形式实现。首先状态 0 发出 trigger 信号，转到状态 1，等待 echo 信号上升沿。当 echo 信号上升沿到达时，将计数器值置为 0，进入状态 2，等待 echo 信号下降沿，当 echo 信号下降沿到达时，读计数器值并计算距离，同时调用 EINK 显示函数进行显示，并通过串口将距离信息发送给计算机。另外，当距离过近时（小于 20cm），红色 led 灯会被点亮。

主逻辑代码如下：

```
1.  void disTask()
2.  {
3.      char echo_read;
4.      char state=0;
5.      while(1)
6.      {
7.          if(state == 0) //发出 trigger 信号
```

```
        {
            Cy_GPIO_Write(dis_trig_PORT, dis_trig_NUM, 1);
            vTaskDelay(1);
            Cy_GPIO_Write(dis_trig_PORT, dis_trig_NUM, 0);

            state = 1;

        }
        else if(state == 1) //等待 echo 上升沿
        {
            echo_read = Cy_GPIO_Read(echo_PORT, echo_NUM);
            if(echo_read==1UL)
            {
                //printf("up side\r\n");
                state = 2;
                Counter_1_SetCounter(0);
            }
        }
        else if(state == 2) //读数
        {
            echo_read = Cy_GPIO_Read(echo_PORT, echo_NUM);
            if(echo_read != 1UL)
            {
                uint32 result = Counter_1_GetCounter();
                double dis = calculateDis(result);

                if(dis>550)
                {
                    printf("out of range!\r\n");
                }
                else
                {
                    printf("distance:%.1f cm\r\n", dis);

                    if(dis<20)
                    {
                        Cy_GPIO_Write(led_PORT, led_NUM, 0);
                    }
                    else
                    {
                        Cy_GPIO_Write(led_PORT, led_NUM, 1);
                    }
                }

```

```
52.                 //xQueueOverwrite(einkQueueHandle,&dis);
53.                 char output[30];
54.                 sprintf(output, "distance:%5.1f cm",dis);
55.                 DisplayText(output);
56.                 vTaskDelay(1000);
57.                 state = 0;
58.             }
59.         }
60.     }
61. }
```

5.4 EINK 显示

主逻辑会在需要时调用墨水屏显示函数，由于墨水屏显示函数还需要当前屏幕上的信息，所以创建两个数组。

```
//one for currentFrame, one for nextFrame
cy_eink_frame_t textPageBackgroundGroup[2][CY_EINK_IMAGE_SIZE];
```

一个存储当前屏幕上的信息，另一个存储欲显示的信息，显示函数部分如下：

```
static uint8_t curframe = 0;

cy_eink_frame_t* frame;
if(curframe==0)
{
    frame = textPageBackgroundGroup[1];
}
else
{
    frame = textPageBackgroundGroup[0];
}

uint8_t textOrigin[2] =  {0x00u, 0x00u};
/* Load the frame buffer with the current text Page content */
Cy_EINK_TextToFrameBuffer(frame, text,
                          CY_EINK_FONT_16X16BLACK, (uint8_t*) textOrigin);

/* Perform a full update to avoid ghosting as the text pages differ
   significantly from one another and also from the main menu images */
Cy_EINK_ShowFrame(textPageBackgroundGroup[curframe], frame,
                  CY_EINK_FULL_2STAGE, CY_EINK_POWER_AUTO);
```

6. 实验结果

达到了实验内容的要求，利用超声波模块测量距离，将距离信息显示在 EINK 墨水屏上，同时通过串口发送给电脑端。当距离过近时（小于 20cm），触发警报，亮红灯。

然而由于 EINK 刷新需要一定时间，所以其显示有大概 0.5s 的延迟。

7. 实验中遇到的问题与解决方法

7.1 Echo 引脚的输入问题

在实验过程中我发现 Echo 引脚输入无法正确检测，结果一直为低电平。于是我用示波器单独检查了超声模块上的 Trig 信号与 Echo 信号，发现二者均正常工作。我猜测可能是引脚驱动模式的问题，于是我将超声模块上的 Echo 信号与 Psoc 上的引脚连接起来，再用示波器观测，发现 Echo 信号会有一个小幅上升，但远没有达到 Psoc 中高电平的要求，猜想得到验证。所以我将 Psoc 原理图中 Echo 信号的驱动模式从 strong drive 改为 resistive pull up，问题解决。

7.2 freeRTOS 多任务问题

刚开始，我用 freeRTOS 将 eink 显示与超声波测距安排在两个任务中，二者之间用 xQueue 通讯。但这样安排会出问题，程序会卡死，我猜测可能是由于超声测距任务并不时时刻刻都在向 eink 显示任务发送消息，我仔细考虑了一下，超声测距任务需要捕捉 echo 信号的上升沿与下降沿，如果同时有另一个任务运行，很可能捕捉不到或不准确，所以我决定使 eink 显示任务只负责初始化，显示函数还是在超声测距任务中调用，这样就可以解决问题，同时提高测量精度。

7.3 PSOC 读取 pin 值问题

Echo 信号的输入需要读取 GPIO 引脚值，除了上述驱动模式的问题外，我发现有些引脚无法输入或输出。

8. 体会、收获与建议

这次实验实践了用 Psoc 设计一个实用系统的过程，本以为有了提高实验的基础，这个实验会比较顺利，但实际上还是遇到了一些问题，在解决问题的过程中我学到了很多，同时也为创新实验打下了基础，我认为为这次实验付出的努力是值得的。