

2021年研究生专业实训报告



**实训课题 ：物联网智能硬件设计开发课程**

姓名：羊俊华

学号：2112003321

学院：信息工程学院

指导老师：顾东袁

时间：2021年4月9日~2021年6月10日

报告提交日期：2021年06月10日

目录

[物联网智能硬件设计开发课程 实训报告 1](#_Toc25114)

[第1章 CO2传感器调试 1](#_Toc26392)

[1.1 输入输出中断捕获程序实现 1](#_Toc2068)

[1.2 在主函数的循环中检测脉宽 2](#_Toc25965)

[第2章 综合功能编写及调试 3](#_Toc23531)

[2.1 库函数的应用 3](#_Toc7814)

[2.2 主要逻辑实现 4](#_Toc12130)

[物联网智能硬件实训课程个人验收表 1](#_Toc20980)

物联网智能硬件设计开发课程 实训报告

# CO2传感器调试

1.1 输入输出中断捕获程序实现

首先初始化定时器频率，修改分频值可以修改定时器计数频率。

F = 主频 / ((计数值+1)\*(预分频器+1))

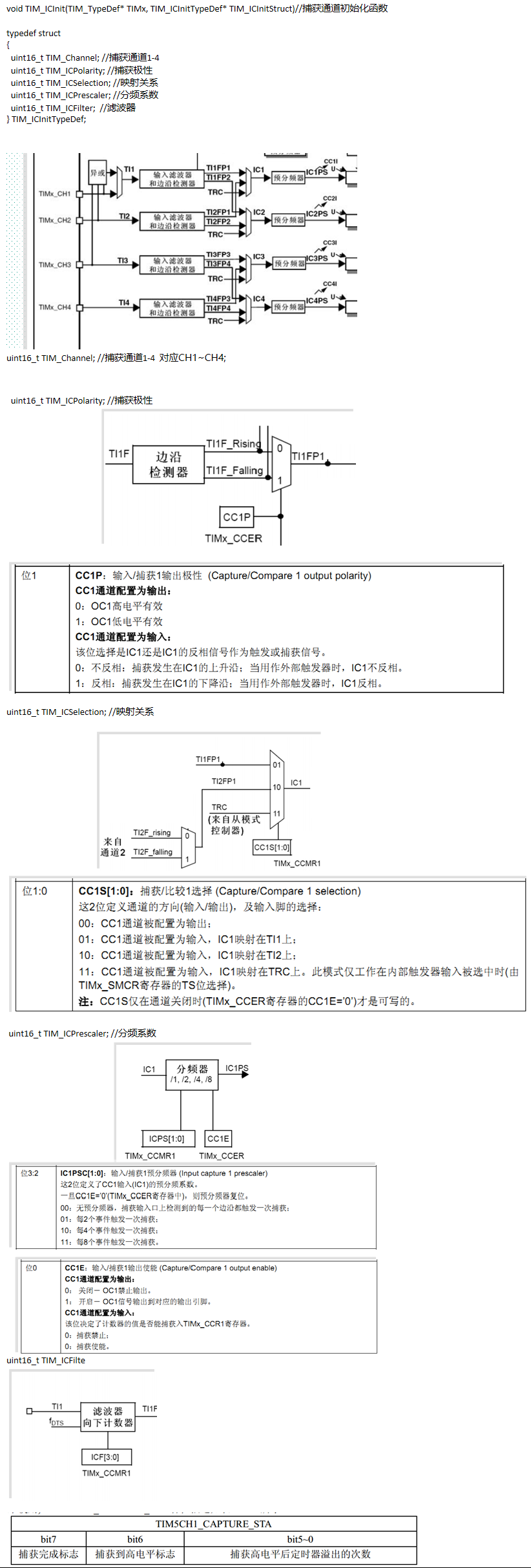
1. TIM5\_CH3\_Cap\_Init(0XFFFFFFFF,90-1); //以1MHZ的频率计数

计数器值为0xFFFFFFFF，预分频值为90，以90M/90=1MHz计数，其捕获时间精度为1us。

在定时器中定义了

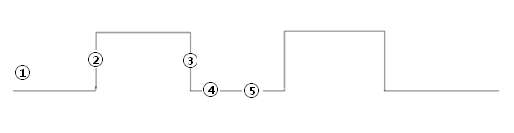
1. u8  TIM5CH3\_CAPTURE\_STA;//输入捕获状态
2. u32 TIM5CH3\_CAPTURE\_VAL;//输入捕获值

其中TIM5CH3\_CAPTURE\_STA作为捕获中的标志状态，定义如下：



接下来是定时器的中断服务程序，主要获得PA2的上升沿电平时间。

1. **void** HAL\_TIM\_PeriodElapsedCallback(TIM\_HandleTypeDef \*htim)//更新中断（溢出）发生时执行
2. {
3. **if**((TIM5CH3\_CAPTURE\_STA&0X80)==0)//还未成功捕获
4. {
5. **if**(TIM5CH3\_CAPTURE\_STA&0X40)//已经捕获到高电平了
6. {
7. **if**((TIM5CH3\_CAPTURE\_STA&0X3F)==0X3F)//高电平太长了
8. {
9. TIM5CH3\_CAPTURE\_STA|=0X80;      //标记成功捕获了一次
10. TIM5CH3\_CAPTURE\_VAL=0XFFFFFFFF;
11. }**else** TIM5CH3\_CAPTURE\_STA++;
12. }
13. }
14. }



①->②段，产生上升沿，出现捕获事件中断，TIM3CH2\_CAPTURE\_STA的6位置1，标志着已经捕获到了上升沿，将参数清0，将CNT计数器置0，然后将捕获类型改设置为下降沿捕获，为了等待③下降沿；

②->③段，这是个下降沿，产生了捕获事件中断，将TIM3CH2\_CAPTURE\_STA的7位置1，标志成功完成了一次捕获上升沿和下降沿，将CNT计数器值赋给TIM3CH2\_CAPTURE\_VAL，这时已经完成一次测脉宽。然后将捕获类型改设置为上升沿捕获。

如果过长时间没有捕获到下降沿，那么就将当前TIM3CH2\_CAPTURE\_VAL设置为当前总的计数次数0xFFFFFFFF。

1.2 在主函数的循环中检测脉宽

正如上面所述的TIM3CH2\_CAPTURE\_STA的格式，先判断第7位是否捕获到一次高低电平，后六位是溢出次数，乘以计数总次数0xFFFFFFFF就是总的溢出时间，最后加上TIM3CH2\_CAPTURE\_VAL的一次高低电平的持续时间，并求出占空比，转化为0~2000ppm的CO2的浓度，最后置零TIM3CH2\_CAPTURE\_STA，以作下一次捕获的准备。

1. **if**(TIM5CH3\_CAPTURE\_STA&0X80)        //成功捕获到了一次高电平
2. {
3. temp=TIM5CH3\_CAPTURE\_STA&0X3F;
4. temp\*=0XFFFFFFFF;               //溢出时间总和
5. temp+=TIM5CH3\_CAPTURE\_VAL;      //得到总的高电平时间
6. printf("HIGH:%lld us\r\n",temp);//打印总的高点平时间
7. **long** **long** a=(temp\*1.00/10)\*20;
8. TIM5CH3\_CAPTURE\_STA=0;          //开启下一次捕获
9. }

# 综合功能编写及调试

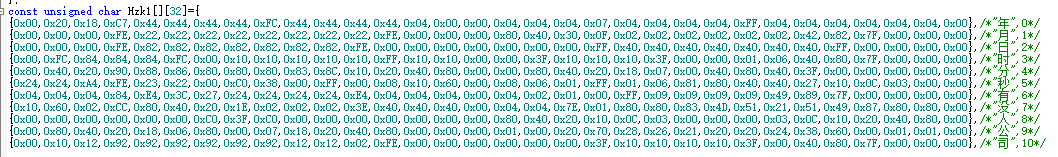
* 1. 库函数的应用

注：综合代码中部分传感器代码库函数：Esp8266代码是田晓鹏完成；打印OLED代码、获取RTC时间数据由傅金波完成；热释电红外传感器代码由刘利松调试完成。综合功能中用到了这些库函数，所以在这里写一下。

* + 1. 变量和宏定义

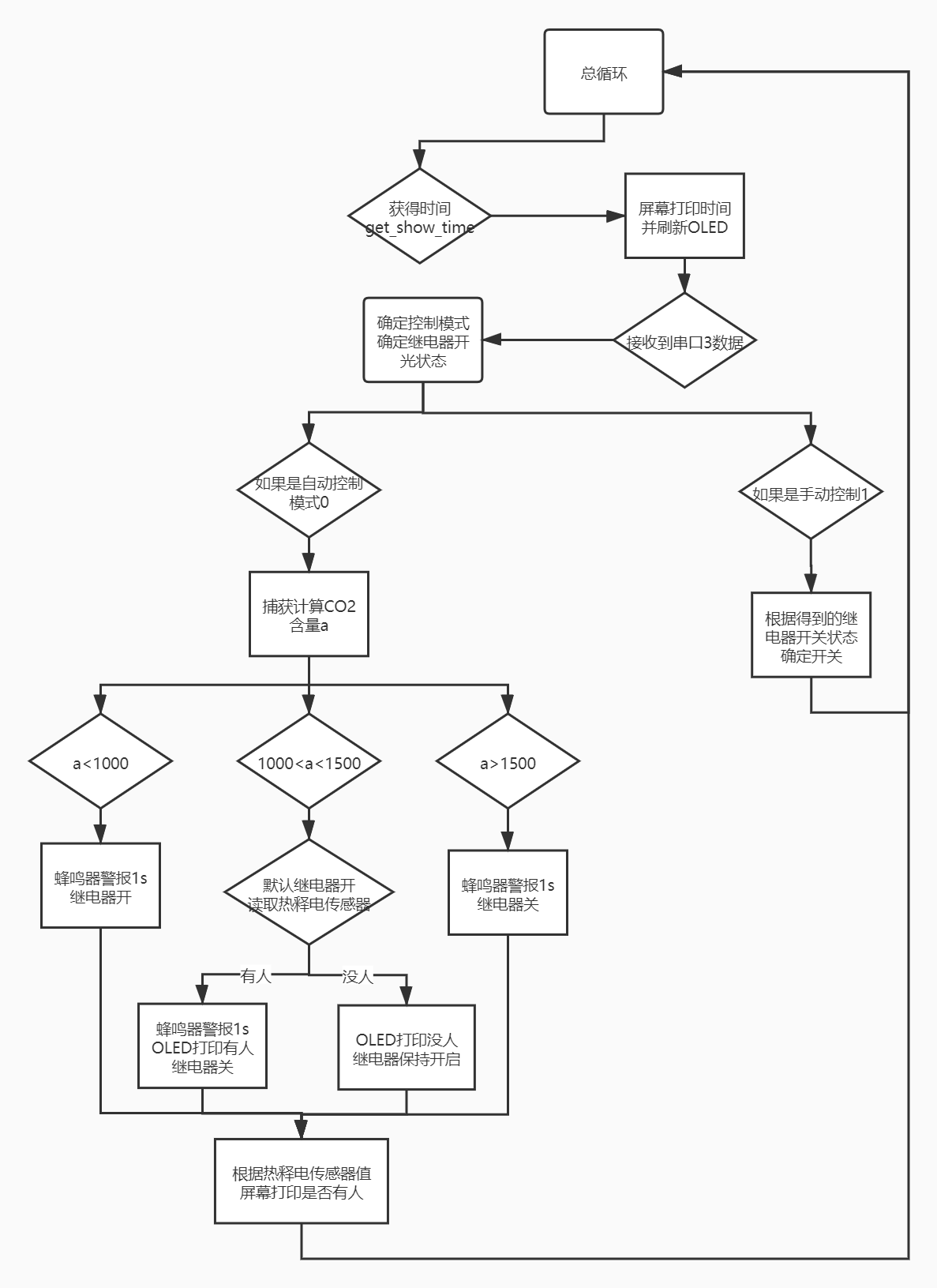
1. u8 t=0;/\*DS3231计时\*/
2. u8 j =0;/\*阿里云从串口3输入的继电器开关状态\*/
3. u8 mode=0;/\*阿里云从串口3输入的控制模式\*/
4. **long** **long** temp=0;/\*CO2传感器上升沿占空比\*/
5. **long** **long** a=0;/\*二氧化碳含量\*/
6. **int** flag = 0x0F;/\*作为一个4位的标志位，对0-1000，1000-1500，1500-的三个范围限定蜂鸣器1s的蜂鸣\*/
7. u16 xx;/\*收到由串口得到手动控制的继电器开关值\*/
8. /\*#define Beep PEout(15),蜂鸣器输出\*/
9. Beep=1;
10. /\*#define JDQ PEout(12),继电器输出\*/
11. JDQ=1;
    * 1. 初始化传感器
12. /\*初始化USART，与PC机通信\*/
13. uart\_init(115200);
14. /\*初始化串口3波特率为115200，串口3与8266通信\*/
15. usart3\_init(115200);
16. /\*初始化OLED，PB10(SCL),PB11(SDA),PE14(RES)\*/
17. OLED\_Init();
18. /\*初始化继电器蜂鸣器,PE12(继电器),PE15(蜂鸣器)\*/
19. BeepJDQ\_Init();
20. /\*初始化热释电传感器，PE3(热释电) \*/
21. PIR\_Init();
22. /\*初始化DS3231,首次设置时间，设置完成后注释掉，重新下载一次程序即可掉电保存\*/
23. DS3231\_Init();
    * 1. 库函数
24. /\*x,y:起点坐标 num:显示数字 len:数字位数 size:字体 mode:0,反色显示;1,正常显示\*/
25. OLED\_ShowNum(0,27,calendar.hour,2,16,1);
26. /\*x,y:起点坐标 size1:字体大小 \*chr:字符串起始地址 mode:0,反色显示;1,正常显示\*/
27. OLED\_ShowString(17,27,":",16,1);
28. /\*x,y:起点坐标 num:汉字对应的序号 mode:0,反色显示;1,正常显示\*/
29. OLED\_ShowChinese(30,48,0,16,1);

num所对应的序号是汉字库Hzk1的下标(将所需要汉字转化为二进制)



* 1. 主要逻辑实现

程序逻辑图：



每次接收到8266传过来的模式和继电器开关值，若是模式0，则进入自动模式，当捕获的CO2浓度小于1000ppm的时候，蜂鸣器报警1s，继电器为开，风扇开，等于向内通入CO2；

当CO2浓度大于1500ppm的是否，蜂鸣器报警1s，继电器为关，风扇关，等于维持CO2浓度；

当CO2浓度小于1500ppm，且大于1000ppm的时候，继电器和风扇默认为开，同时判断是否有人，若有人，则关继电器和风扇关，且报警1s，若没人，则继电器和风扇保持开启状态。

1. **while**(1)
2. {
3. /\*串口接受一次数据\*/
4. **if**(USART3\_RX\_STA&0X8000)
5. {
6. /\*xx作为标志位，得到从阿里云传入的功能码 48 49 50 51 即为功能码0 1 2 3，表示为继电器开关状态0 1，模式：自动模式0 手动模式1 \*/
7. xx =(**int**)USART3\_RX\_BUF[0];
8. **if** (xx==50)
9. mode=0;
10. **else** **if**(xx==51)
11. mode=1;
12. **if** (xx==48)
13. j=0;
14. **else** **if**(xx==49)
15. j=1;
16. }
17. /\*开启自动模式\*/
18. **if** (mode==0)
19. {
20. /\*当CO2含量a小于1000并且进入标志位为1，蜂鸣器响1s，继电器开，并复位进入标志位\*/
21. **if** (a<1000&&(flag&0x01)==1)
22. {
23. Beep=1;
24. JDQ=1;
25. flag=0x0E;
26. delay\_ms(1000);
27. Beep=0;
28. }
29. /\*当CO2含量a大于1500并且进入标志位为1，蜂鸣器响1s，继电器关，并复位进入标志位\*/
30. **else** **if** (a>1500&&(flag&0x02)==2)
31. {
32. Beep=1;
33. JDQ=0;
34. flag = 0x0D;
35. delay\_ms(1000);
36. Beep=0;
37. }
39. /\*当CO2含量a大于1000并小于1500，继电器开，如果检测到有人，蜂鸣器报警0.1s，OLED打印有人信息，继电器关闭
40. \* 如果检测到没人,OLED打印没人信息，继电器继续开启
41. \*/
42. **if** (a<1500&&a>1000)
43. {
44. JDQ=1;
45. **if** (PIR\_Scan()==1)
46. {
47. Beep=1;
48. JDQ=0;
49. delay\_ms(100);
50. Beep=0;
51. }
52. **else** **if** (PIR\_Scan()==0)
53. {
54. JDQ=1;
55. }
56. }
57. /\*在整个自动模式中都会检测有没有人，OLED打印相关信息\*/
58. **if** (PIR\_Scan()==1){}
59. **else** **if** (PIR\_Scan()==0)  {}
60. }
61. /\*手动模式 得到继电器输入值，并在OLED上面打印\*/
62. **else** **if** (mode==1)
63. {
64. **if**(j==0)JDQ=0;
65. **else** **if**(j==1)JDQ=1;
66. }
67. }

物联网智能硬件实训课程个人验收表

姓名： 羊俊华 时间： 2021 年 6 月 7 日

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 序号 | 工作内容点 | 完成情况 | 验收情况 | |
| 本人 |  |
|  | 焊接底板 | 完成焊接 | 羊俊华 |  |
|  | CO2传感器软件编写 | 利用输入输出捕获高电平时间，计算占空比，以及CO2浓度 | 羊俊华 |  |
|  | CO2传感器调试 | 完成调试 | 羊俊华 |  |
|  | 传感器硬件功能实现 | 结合CO2传感器和热释电传感器来实现继电器的开关状态的逻辑控制，并打印到屏幕 | 羊俊华 |  |
|  | 综合功能调试 | 完成调试 | 羊俊华 |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

验收说明：

1. 验收资料所在Git目录: T0102/第三部分(焊接调试)/软件调试/CO2读取测试

T0102/第四部分(程序)/综合功能测试

1. 其它事宜说明：综合代码中部分传感器代码库函数：Esp8266代码是田晓鹏完成、打印OLED代码、获取RTC时间数据由傅金波完成、热释电红外传感器代码由刘利松调试完成
2. 请将此电子稿填写完毕后上传到git，同时准备一份打印稿，供验收时使用。并将此表合并入最终实训报告中。