

负压电磁智能车设计方案

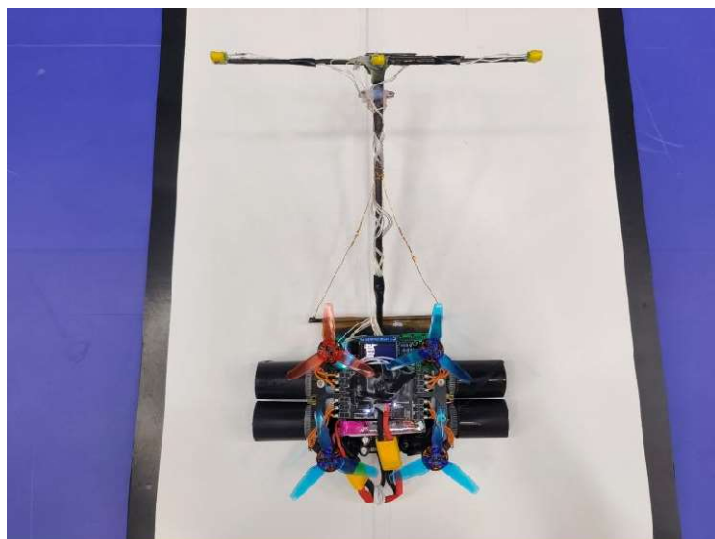
山东大学 基极向上队

1、整体结构

车模采用了双驱动电机、四轮差速结构。车模的整体设计思路来自日本电子鼠，而其中部分细节来自于 X-ray X1 的车模结构。因为规则允许加入负压，所以参考了日本电子鼠的大致思路，进行了车模的降低重心、减重以及使用四个无刷风扇提供向下压力。

由于我们对竞赛整体的经验不足，对于车模的整体结构理解也十分有限，因此，我们在开始设计车模时采用仿真的方式进行了一些大概结构的确定。

在此之后的许多细节，由于理论和实际存在差距，我们通过制造并设计实验验证比较，进行不断的迭代，例如单风扇抽气制造负压以及四风扇提供下压力之间的对比；C 车模、D 车模以及自制车模之间结构的对比；车模的齿轮部分，关于齿轮的齿数、模数、传动比等参数；风扇的位置等，通过实践进行方案的验证、对比，不断的迭代车模。



2、底板

在选定这个方案后，开始的思路便是将车模的 PCB 版作为车模的底板，由于体积，位置，高度等限制，PCB 板选择主驱一体的方案，将核心板、母板、运放、驱动集成在一块板上，大幅度减小了整个板子的体积。但是由于 PCB 板的强度问题，会使前瞻有所摇晃，不利于巡线以及元素的判断。在后期更改了方案，将底

板更换为碳纤维板进行支撑，将较薄的 PCB 板位于其上，但是由于选用的 PCB 板较薄、易变形，我们又改为多层碳板将 PCB 平整且牢固的固定在碳板上面。



3、顶板

利用 3D 打印的固定件将四个无刷电机固定在碳板上，四个无刷电调以及其拼装的分电板，置于碳板上，通过此方法能对于底板的强度进行进一步的加固。分体电调相比四合一电调虽然牺牲了体积优势，但有更灵活的装配方式，有助于车模的调整，并尽可能的减小成本。

4、轮胎

设计之初的想法便是使用硅胶轮胎，开始使用了淘宝定制的硅胶轮胎，但是由于我们的车模整体精度要讲求较高，并且有些打印和制作的精度出现问题，我们便自己尝试，学会了制作硅胶轮胎，精度方面有所改善。



A black, cylindrical, hollow object, possibly a container or a component, lying on a light-colored surface. The object has a smooth, reflective surface and a slightly flared top edge.

MCU

MCU Pin List:

Pin	Function	Value
1	P5.1/TxD0_2/LIN_TX_2	33
2	P0.5/AD0/ADCL7/CLKO	35
3	P0.5/AD0/ADCL7/CLKO	35
4	P0.5/AD0/ADCL7/CLKO	35
5	P0.5/AD0/ADCL7/CLKO	35
6	P0.5/AD0/ADCL7/CLKO	35
7	P0.5/AD0/ADCL7/CLKO	35
8	P0.5/AD0/ADCL7/CLKO	35
9	P0.5/AD0/ADCL7/CLKO	35
10	P0.5/AD0/ADCL7/CLKO	35
11	P0.5/AD0/ADCL7/CLKO	35
12	P0.5/AD0/ADCL7/CLKO	35
13	P0.5/AD0/ADCL7/CLKO	35
14	P0.5/AD0/ADCL7/CLKO	35
15	P0.5/AD0/ADCL7/CLKO	35
16	P0.5/AD0/ADCL7/CLKO	35
17	P0.5/AD0/ADCL7/CLKO	35
18	P0.5/AD0/ADCL7/CLKO	35
19	P0.5/AD0/ADCL7/CLKO	35
20	P0.5/AD0/ADCL7/CLKO	35
21	P0.5/AD0/ADCL7/CLKO	35
22	P0.5/AD0/ADCL7/CLKO	35
23	P0.5/AD0/ADCL7/CLKO	35
24	P0.5/AD0/ADCL7/CLKO	35
25	P0.5/AD0/ADCL7/CLKO	35
26	P0.5/AD0/ADCL7/CLKO	35
27	P0.5/AD0/ADCL7/CLKO	35
28	P0.5/AD0/ADCL7/CLKO	35
29	P0.5/AD0/ADCL7/CLKO	35
30	P0.5/AD0/ADCL7/CLKO	35
31	P0.5/AD0/ADCL7/CLKO	35
32	P0.5/AD0/ADCL7/CLKO	35
33	P0.5/AD0/ADCL7/CLKO	35
34	P0.5/AD0/ADCL7/CLKO	35
35	P0.5/AD0/ADCL7/CLKO	35
36	P0.5/AD0/ADCL7/CLKO	35
37	P0.5/AD0/ADCL7/CLKO	35
38	P0.5/AD0/ADCL7/CLKO	35
39	P0.5/AD0/ADCL7/CLKO	35
40	P0.5/AD0/ADCL7/CLKO	35
41	P0.5/AD0/ADCL7/CLKO	35
42	P0.5/AD0/ADCL7/CLKO	35
43	P0.5/AD0/ADCL7/CLKO	35
44	P0.5/AD0/ADCL7/CLKO	35
45	P0.5/AD0/ADCL7/CLKO	35
46	P0.5/AD0/ADCL7/CLKO	35
47	P0.5/AD0/ADCL7/CLKO	35
48	P0.5/AD0/ADCL7/CLKO	35
49	P0.5/AD0/ADCL7/CLKO	35
50	P0.5/AD0/ADCL7/CLKO	35
51	P0.5/AD0/ADCL7/CLKO	35
52	P0.5/AD0/ADCL7/CLKO	35
53	P0.5/AD0/ADCL7/CLKO	35
54	P0.5/AD0/ADCL7/CLKO	35
55	P0.5/AD0/ADCL7/CLKO	35
56	P0.5/AD0/ADCL7/CLKO	35
57	P0.5/AD0/ADCL7/CLKO	35
58	P0.5/AD0/ADCL7/CLKO	35
59	P0.5/AD0/ADCL7/CLKO	35
60	P0.5/AD0/ADCL7/CLKO	35
61	P0.5/AD0/ADCL7/CLKO	35
62	P0.5/AD0/ADCL7/CLKO	35
63	P0.5/AD0/ADCL7/CLKO	35
64	P0.5/AD0/ADCL7/CLKO	35
65	P0.5/AD0/ADCL7/CLKO	35
66	P0.5/AD0/ADCL7/CLKO	35
67	P0.5/AD0/ADCL7/CLKO	35
68	P0.5/AD0/ADCL7/CLKO	35
69	P0.5/AD0/ADCL7/CLKO	35
70	P0.5/AD0/ADCL7/CLKO	35
71	P0.5/AD0/ADCL7/CLKO	35
72	P0.5/AD0/ADCL7/CLKO	35
73	P0.5/AD0/ADCL7/CLKO	35
74	P0.5/AD0/ADCL7/CLKO	35
75	P0.5/AD0/ADCL7/CLKO	35
76	P0.5/AD0/ADCL7/CLKO	35
77	P0.5/AD0/ADCL7/CLKO	35
78	P0.5/AD0/ADCL7/CLKO	35
79	P0.5/AD0/ADCL7/CLKO	35
80	P0.5/AD0/ADCL7/CLKO	35
81	P0.5/AD0/ADCL7/CLKO	35
82	P0.5/AD0/ADCL7/CLKO	35
83	P0.5/AD0/ADCL7/CLKO	35
84	P0.5/AD0/ADCL7/CLKO	35
85	P0.5/AD0/ADCL7/CLKO	35
86	P0.5/AD0/ADCL7/CLKO	35
87	P0.5/AD0/ADCL7/CLKO	35
88	P0.5/AD0/ADCL7/CLKO	35
89	P0.5/AD0/ADCL7/CLKO	35
90	P0.5/AD0/ADCL7/CLKO	35
91	P0.5/AD0/ADCL7/CLKO	35
92	P0.5/AD0/ADCL7/CLKO	35
93	P0.5/AD0/ADCL7/CLKO	35
94	P0.5/AD0/ADCL7/CLKO	35
95	P0.5/AD0/ADCL7/CLKO	35
96	P0.5/AD0/ADCL7/CLKO	35
97	P0.5/AD0/ADCL7/CLKO	35
98	P0.5/AD0/ADCL7/CLKO	35
99	P0.5/AD0/ADCL7/CLKO	35
100	P0.5/AD0/ADCL7/CLKO	35

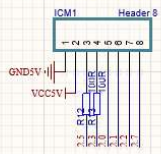
STC14F40K128-64PDS

下载端口

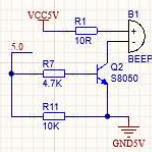
STC14F40K128-64PDS Pin List:

Pin</

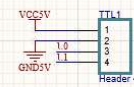
OLED接口



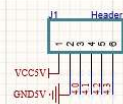
蜂鸣器



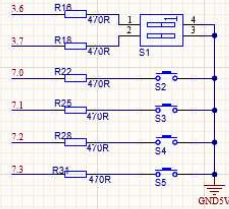
串口接口



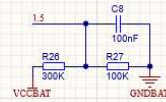
IMU660接口



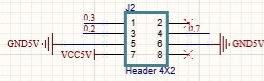
按键



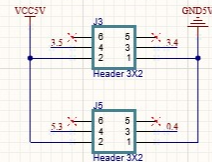
电池电压检测



无线串口



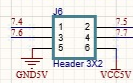
编码器



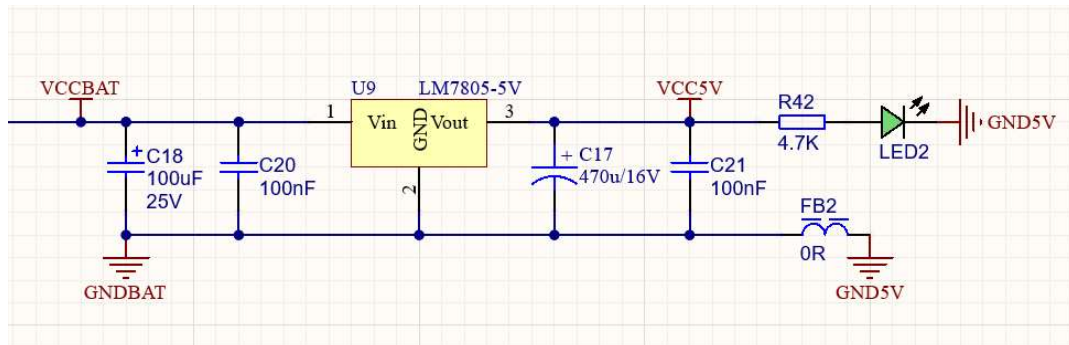
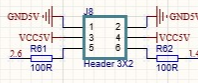
TOF接口

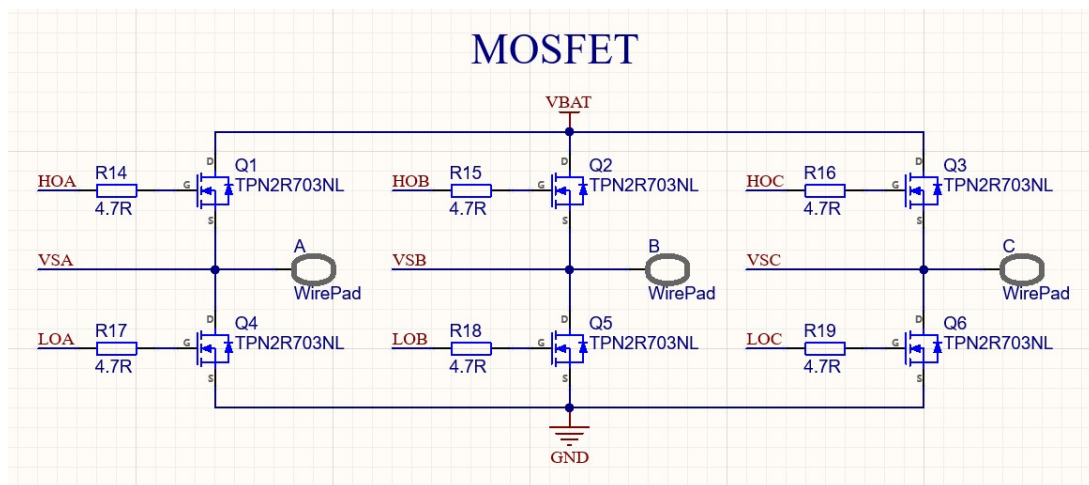
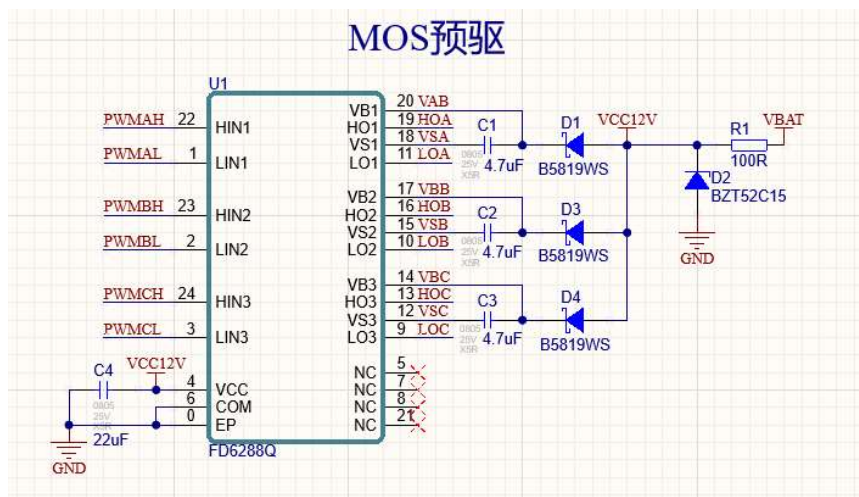
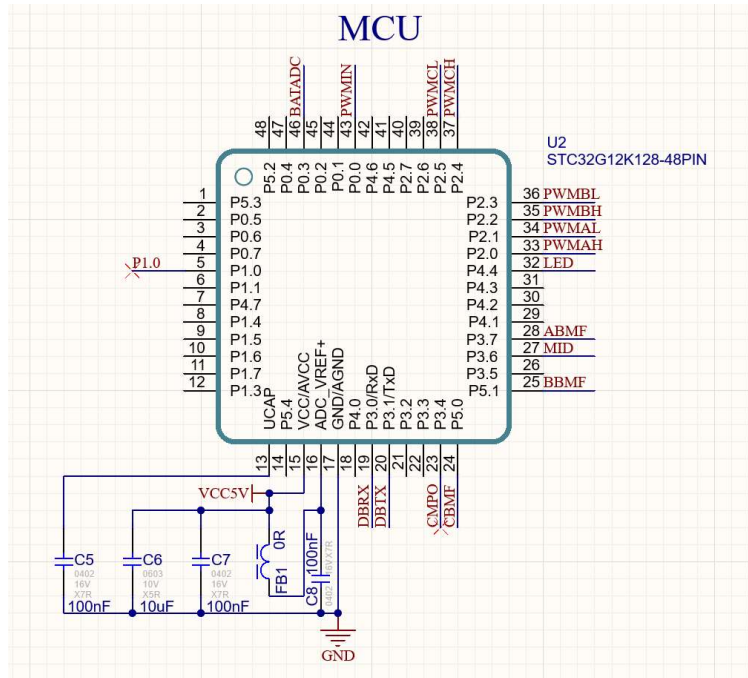


电调信号

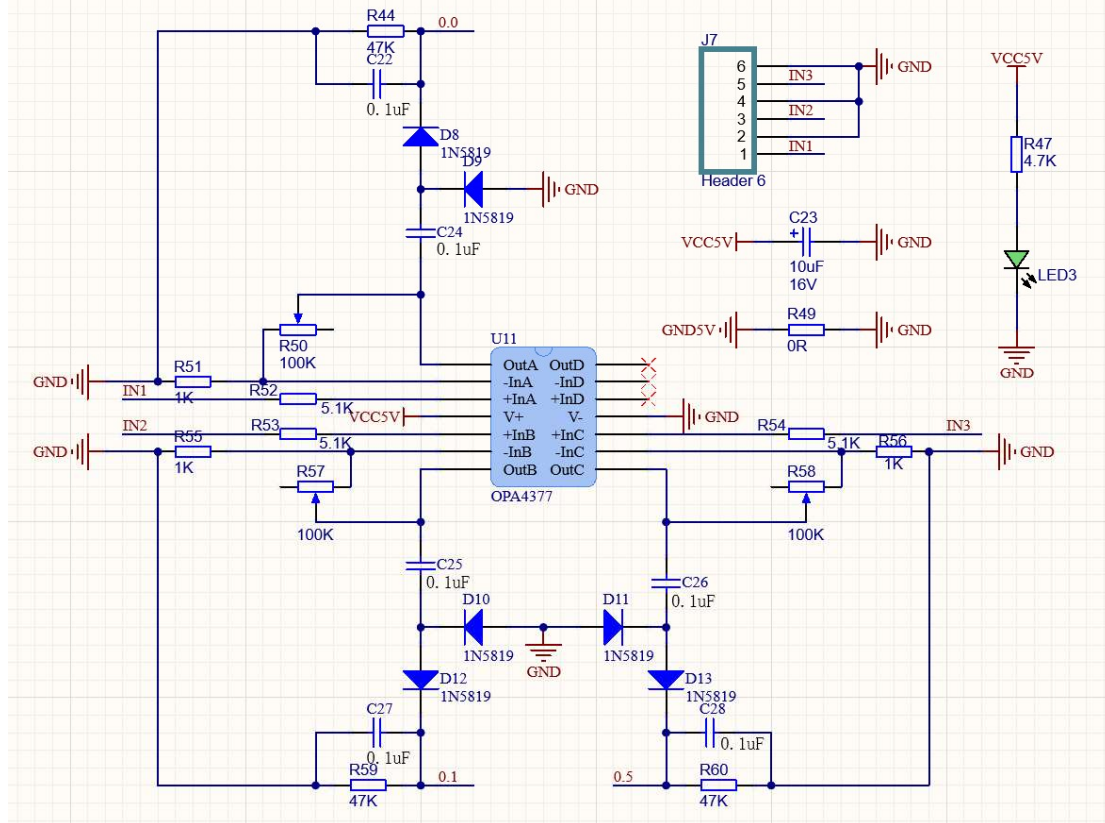


停车触发





电磁运放



6.常用功能函数代码

```
//-----
// 清空采集信号
void CleanADC()
{
    //adc 采样清空为下一次采集做准备
    for(i = 0; i < 3; i++)
    {
        adc_deal[i] = 0;
    }
}
//-----
```

```
//-----
// 获取电磁信号
void GetADC(uint8 flag)
{
    for(i = 0; i < 10; i++)
```

```

    {
        adc_ini[0][i] = adc_once(ADC_P00,ADC_10BIT);
        adc_ini[1][i] = adc_once(ADC_P01,ADC_10BIT);
        adc_ini[2][i] = adc_once(ADC_P05,ADC_10BIT);
    }

    //均值滤波
    for(i = 0; i < 10; i++)
    {
        adc_deal[0] += adc_ini[0][i];
        adc_deal[1] += adc_ini[1][i];
        adc_deal[2] += adc_ini[2][i];
    }

    //归一获取
    for(i = 0; i < 3; i++)
    {
        adc_deal[i] /= 10;

        if(adc_deal[i] > adc_max[i])
            adc_deal[i] = adc_max[i];

        adc_deal[i] = 100 * adc_deal[i] / adc_max[i];
    }

    //屏幕显示
    if(flag)
    {
        ips114_showstr(0, 0, "L : ");
        ips114_showint32(8*10, 0, adc_deal[0],10);
        ips114_showstr(0, 1, "M : ");
        ips114_showint32(8*10, 1, adc_deal[1],10);
        ips114_showstr(0, 2, "R : ");
        ips114_showint32(8*10, 2, adc_deal[2],10);

        CleanADC();
    }
}

//-----
-----

//-----
// 无刷电机输出
void FunPwmOutput()

```



```

{
    pwm_duty(PWMB_CH1_P74, pwm_fun_front);
    pwm_duty(PWMB_CH2_P75, pwm_fun_front);
    pwm_duty(PWMB_CH3_P76, pwm_fun_behind);
    pwm_duty(PWMB_CH4_P77, pwm_fun_behind);
}
//-----

//-----
// 方向环函数（含脱线保护与速度决策）
void DiffPid()
{
    //差比和获取 error
    if(adc_deal[2] >= adc_deal[0])
        error = (200 * (adc_deal[2] - adc_deal[0])) / (adc_deal[2] + adc_deal[0]);
    else
    {
        error = (200 * (adc_deal[0] - adc_deal[2])) / (adc_deal[2] + adc_deal[0]);
        error = -error;
    }

    //正常循迹
    if(adc_deal[0] || adc_deal[2])
    {
//        pid.Kp = (0.02 * error)/100 + 0.06;
        diff = diff_pid_realize(&pid, error);
        pidl.setspeed = setspeed + diff;
        pidr.setspeed = setspeed - diff;
    }
    //脱线制停
    else
        flag_st = 1;
}
//-----

//-----
// 获取编码器数值
void GetEncoder()
{
    //读取采集到的编码器脉冲数

```

```

        pluse_L = ctimer_count_read(SPEEDL_PLUSE);
        pluse_R = ctimer_count_read(SPEEDR_PLUSE);
        //计数器清零
        ctimer_count_clean(SPEEDL_PLUSE);
        ctimer_count_clean(SPEEDR_PLUSE);
        //采集方向信息
        if(SPEEDL_DIR != 1)
        {
            pluse_L = -pluse_L;
        }

        if(SPEEDR_DIR == 1)
        {
            pluse_R = -pluse_R;
        }
        //4us

        if(flag_inte)
            pluse_inte += (pluse_R + pluse_L) / 2;
    }
    //-----

    //-----
    // 速度环函数
    void MotorPid()
    {
        dutyL += mo_pid_realize(&pidl,pluse_L);
        dutyR += mo_pid_realize(&pidr,pluse_R);
    }
    //-----

    //-----
    // 有刷电机输出
    void PwmOutput()
    {
        //输出限幅
        if(dutyL > 9999 )
            dutyL = 9999;
        if(dutyR > 9999)
            dutyR = 9999;
        if(dutyL < -9999 )
            dutyL = -9999;
        if(dutyR < -9999)

```

```

        dutyR = -9999;

//完全停车
if(flag_park >= 2 || flag_st == 1)
{
    dutyL = 0;
    dutyR = 0;
    pwm_fun_front = 1.0 / 20 * 10000;
    pwm_fun_behind = 1.0 / 20 * 10000;
    FunPwmOutput();
    LED = 0;
}

//左轮
if(dutyL >= 0) //正转
{
    pwm_duty(PWMA_CH1P_P60, dutyL);
    pwm_duty(PWMA_CH2P_P62, 0);
}
else //反转
{
    pwm_duty(PWMA_CH1P_P60, 0);
    pwm_duty(PWMA_CH2P_P62, -dutyL);
}

//右轮
if(dutyR >= 0) //正转
{
    pwm_duty(PWMA_CH3P_P64, 0);
    pwm_duty(PWMA_CH4P_P66, dutyR);
}
else //反转
{
    pwm_duty(PWMA_CH3P_P64, -dutyR);
    pwm_duty(PWMA_CH4P_P66, 0);
}
}

//-----
-----

//-----
// 获取电池电压（含低压保护）
//-----
-----

```

```

void GetVoltage()
{
    battery_voltage = 4 * (((uint32)adc_once(BAT_VOL_PIN, ADC_12BIT) * 5000) /
4096);
    /*原理
    ad_result = adc_once(BAT_VOL_PIN, ADC_12BIT);
    temp = (((uint32)ad_result * 5000) / 4096); //计算出当前 adc 引脚的电压 计
算公式为 ad_result*VCC/ADC 分辨率    VCC 单位为 mv
    battery_voltage = temp * 4; //根据引脚电压 和分压电阻的阻值计算电池电
压 计算公式为 引脚电压*(R2+R3)/R3    R3 为接地端电阻
    */
    if(battery_voltage <= 11800)
        BEEP = 1;
    else
        BEEP = 0;
}

```