

# 圆点博士 微型四轴飞行器

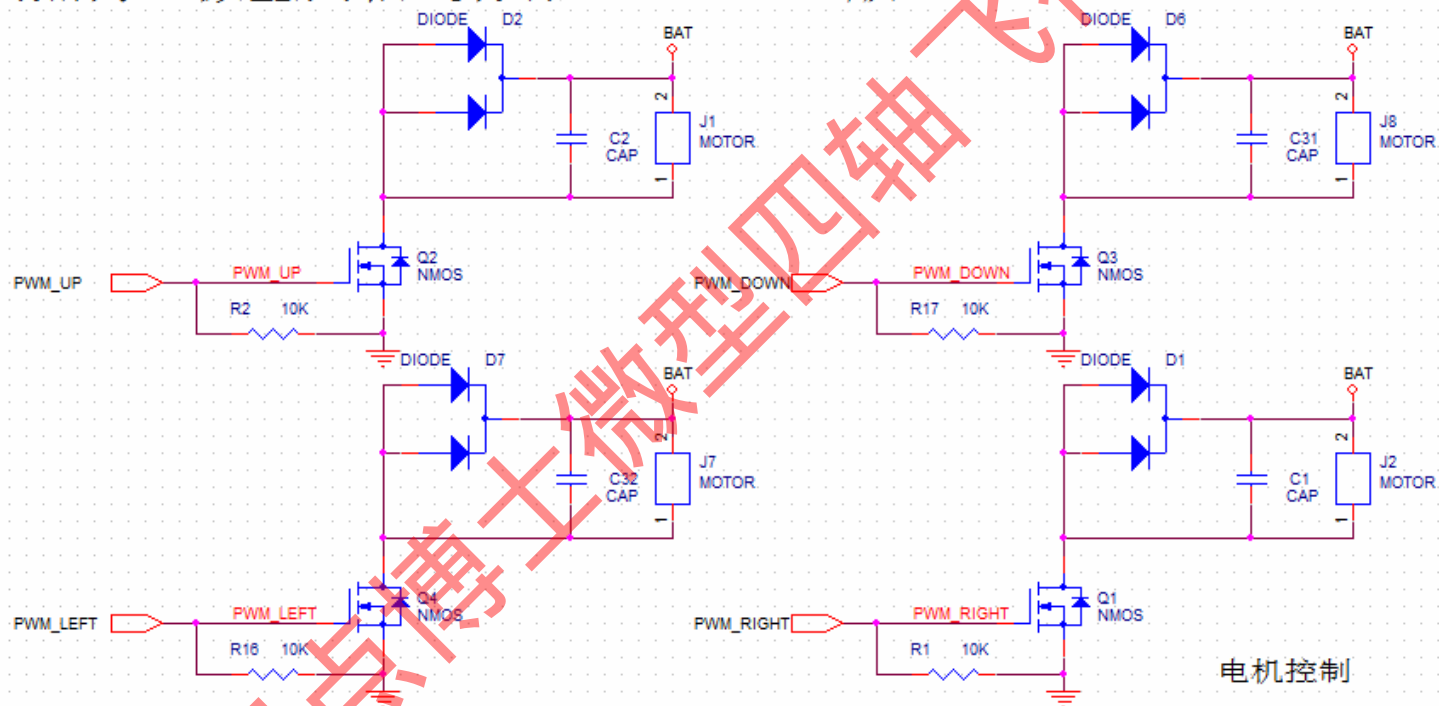
## BS001-B/C版本 使用手册

2012.12.14.01

圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

# 圆点博士微型四轴飞行器原理图（一）

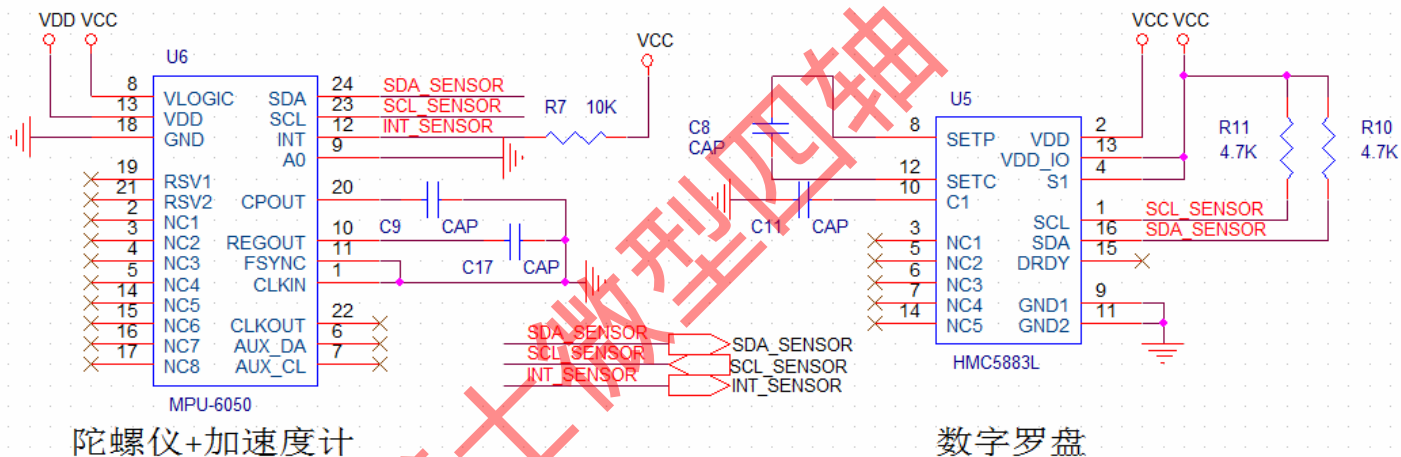
圆点博士微型四轴飞行器： BS001-B版



圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器原理图（二）

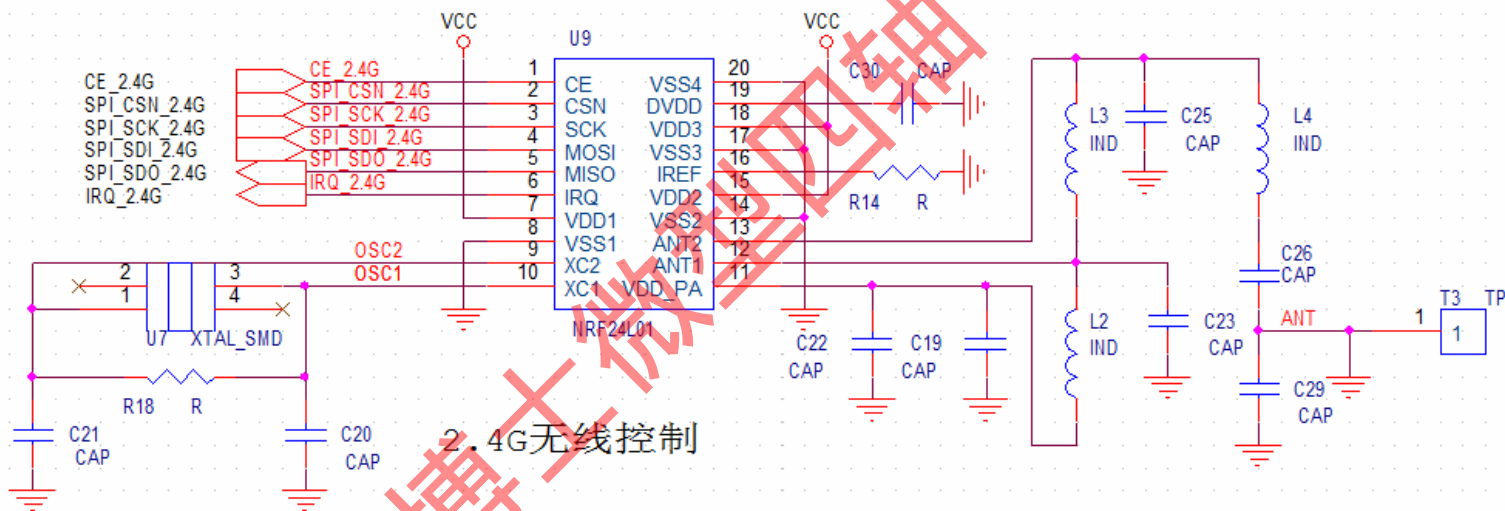
# 圆点博士微型四轴飞行器：BS001-B版



圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器原理图（三）

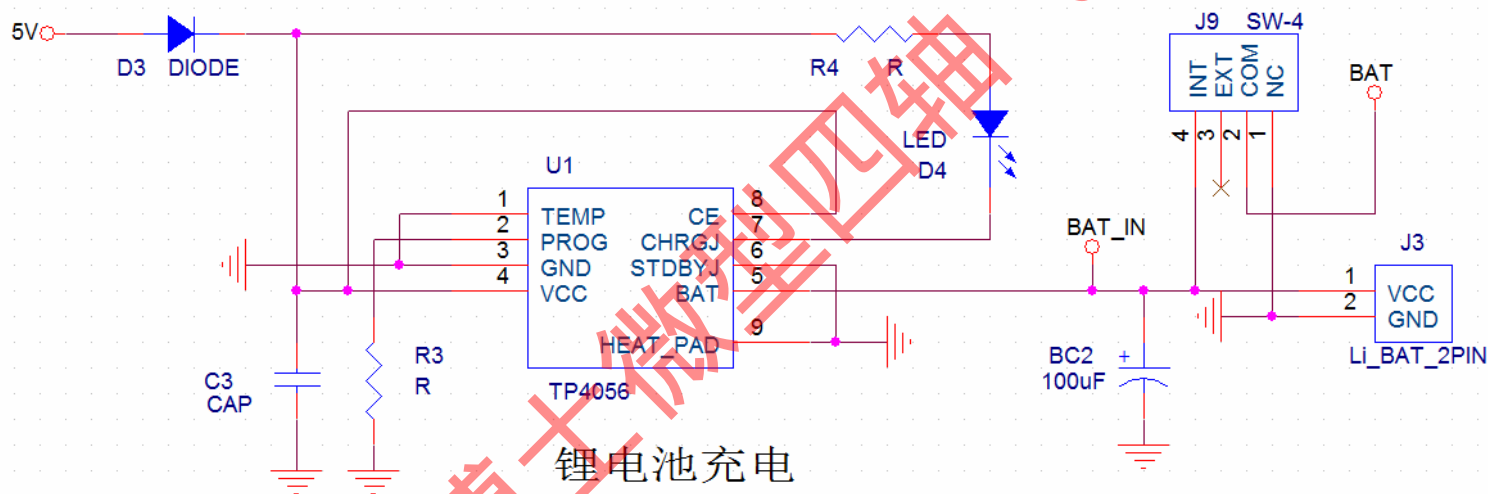
### 圆点博士微型四轴飞行器：BS001-B版



圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器原理图（四）

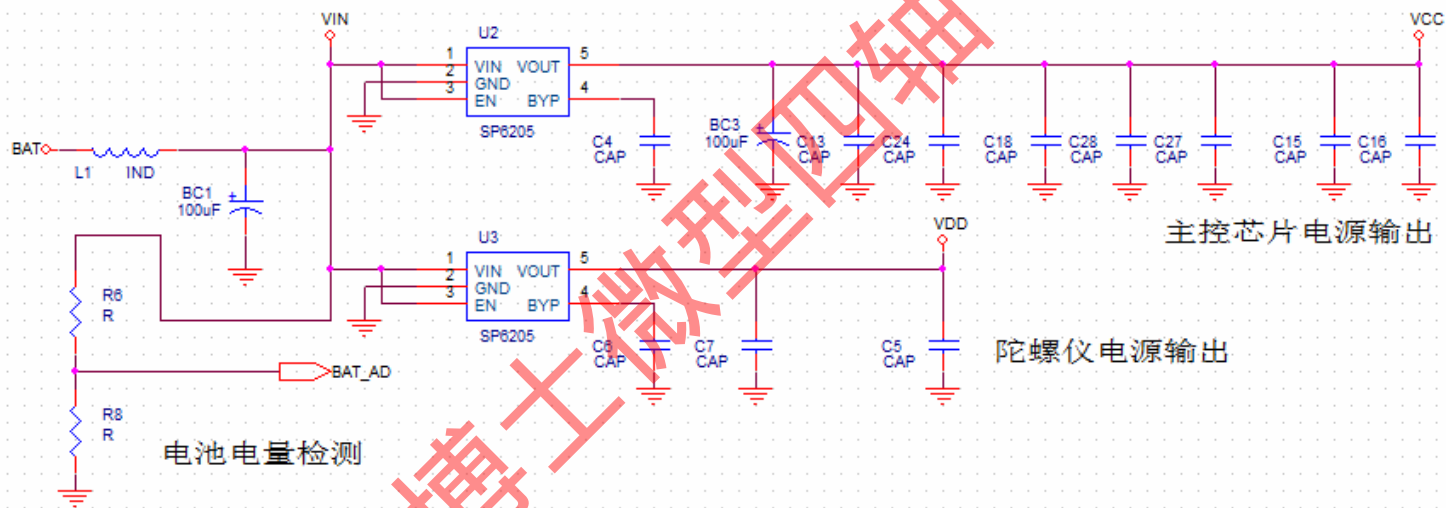
### 圆点博士微型四轴飞行器：BS001-B版



圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器原理图（五）

### 圆点博士微型四轴飞行器：BS001-B版



圆点博士微型四轴飞行器官方网站：<http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器原理图（六）



圆点博士微型四轴飞行器官方网站：<http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器元器件表（一）

圆点博士微型四轴飞行器器件清单			
标号	位置	型号	描述
芯片类			
1	U8	STM32F103T8U6	主控芯片
2	U1	TP4056	充电芯片
3	U2,U3	SP6205	电压转换LDO
4	U6	MPU-6050	陀螺仪+加速度计
5	U5	HMC5883L	磁力仪
6	U9	NRF24L01+	无线芯片
7	Q1,Q2,Q3,Q4	SI2302	电机PWM驱动
8	D1,D2,D6,D7	BAT54C	电机保护二极管
9	D3	SS12	电源反插保护二极管
10	D4,D5	LED	LED指示灯
晶振类			
1	U4,U7	SMD3225 16M	主控芯片使用
电机			
1	J1,J2,J7,J8		电机电源
2	T1,T2,T5,T6		电机+桨

圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>



## 圆点博士微型四轴飞行器元器件表（二）

圆点博士微型四轴飞行器器件清单			
标号	位置	型号	描述
接口			
1	J5	Micro-USB-B	USB接口
2	J3	2脚带座插针	锂电池插座
3	J6	4脚插针	主控芯片程序下载插针
4	J9	开关	用于关闭飞行器电源
5	J4	无（不上）	用于使用液晶屏进行调试
6	U10	无（不上）	用于使用串口进行调试
7	T3	无（不上）	天线
8	T4,T7,T8	无（不上）	测试点

圆点博士微型四轴飞行器官方网站：<http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器元器件表（三）

圆点博士微型四轴飞行器器件清单			
标号	位置	型号	描述
电感			
1	L2	2.7nH	
2	L4	3.9nH	
3	L3	8.2nH	
4	L1	10uH	
电阻			
1	R5	470	471
2	R3,R4	12K	122
3	R13	15K	152
4	R7,R10,R11,R15	22K	222
	R1,R2,R9,R12,R16		
5	R17	10K	103
6	R14	22K	223
7	R6,R8	100K	104
8	R18	1M	105

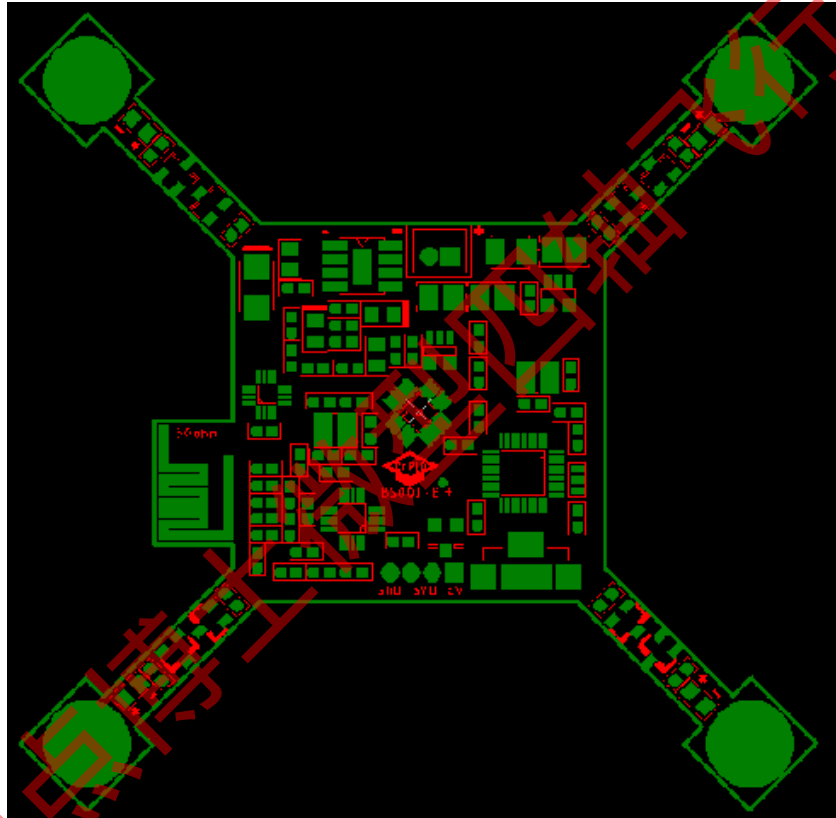
圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器元器件表（四）

圆点博士微型四轴飞行器器件清单			
标号	位置	型号	描述
电容			
1	BC1,BC2,BC3	47uF	
电容			
1	C4,C6	0.01uF	103
2	C1,C2,C3,C5,C13, C15,C16,C17,C18, C24,C27,C28,C31, C32	0.1uF	104
3	C8	0.22uF	224
4	C14	1uF	105
5	C11	4.7uF	475
6	C7	10uF	106
7	C9,C22	2.2nF	222
8	C30	33nF	333
9	C26	1.5pF	
10	C19	4.7pF	
11	C10,C12,C20,C21	22pF	220
12	C23,C25,C29	无（不上）	

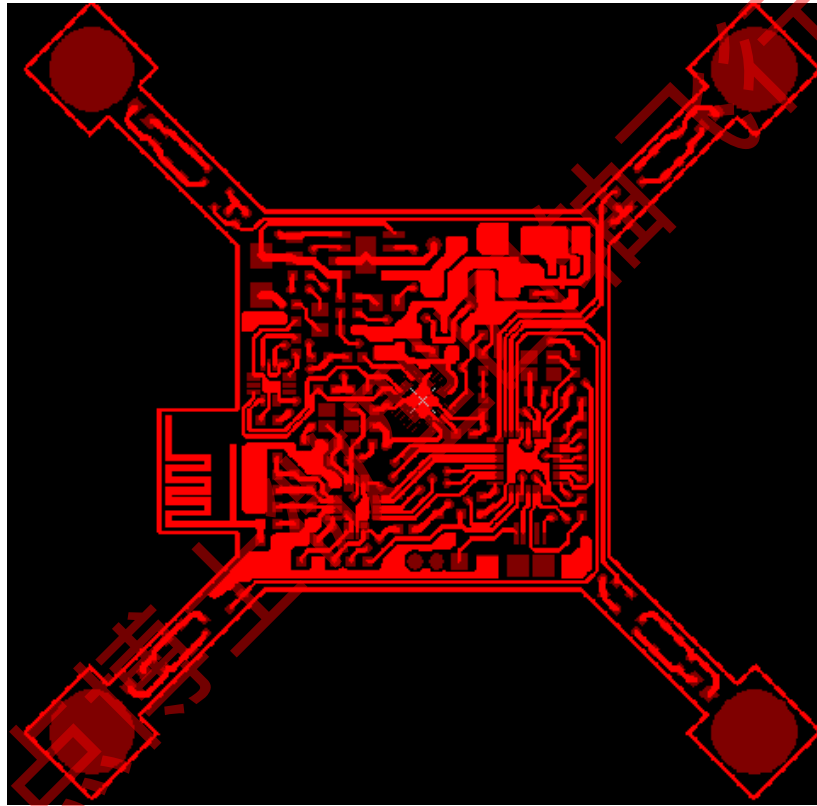
圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器PCB（一）



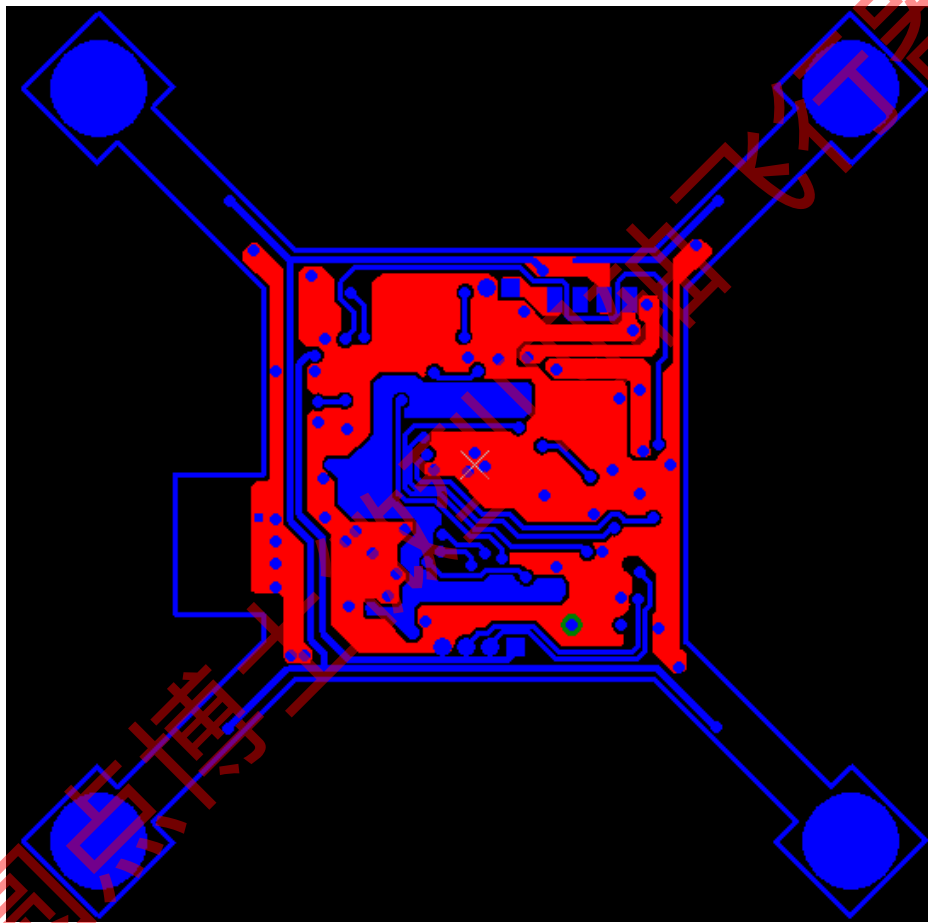
圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器PCB（二）



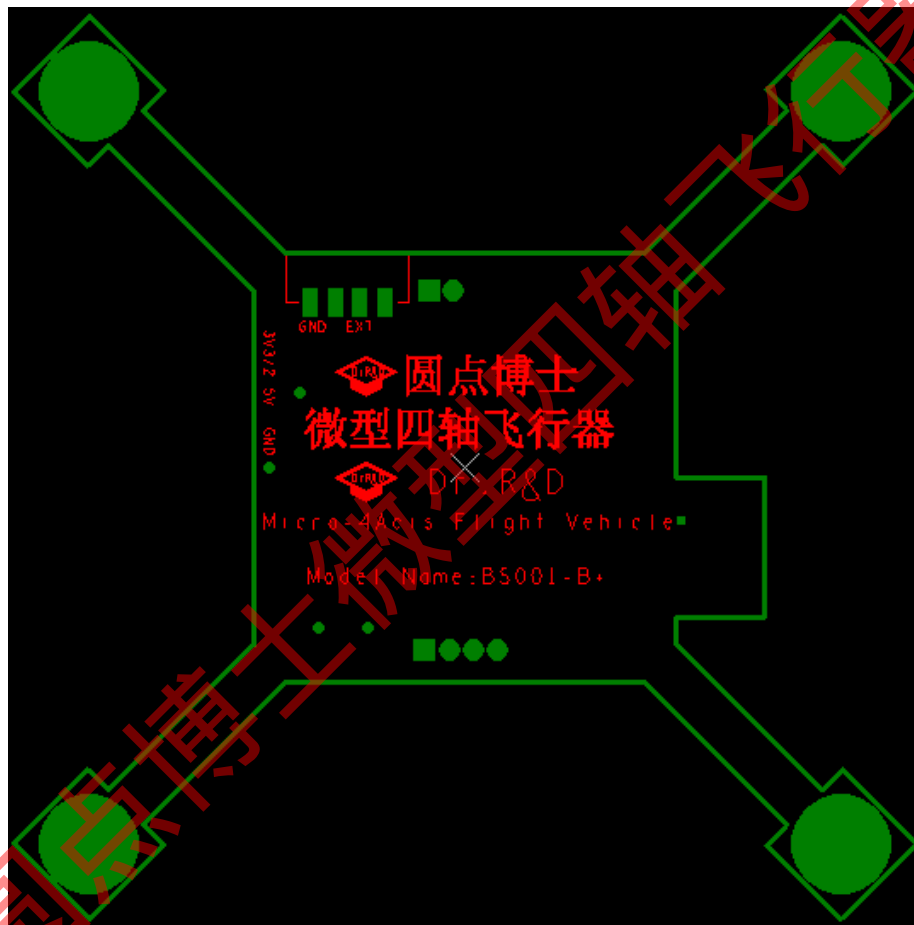
圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器PCB（三）



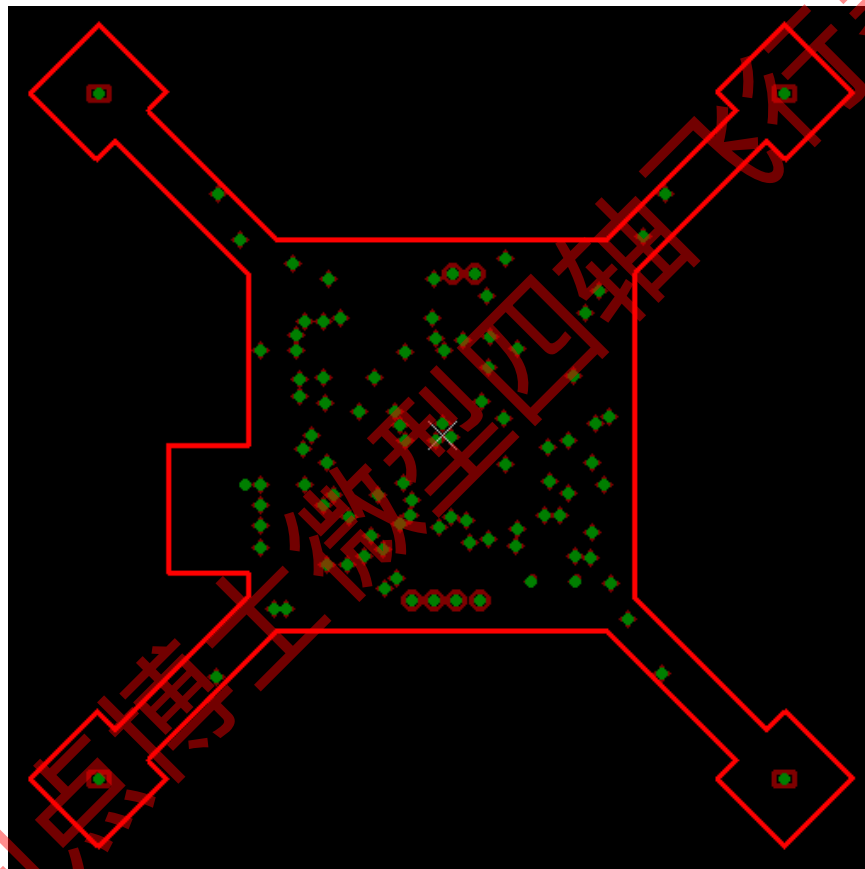
圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器PCB（四）



圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器PCB（五）



圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

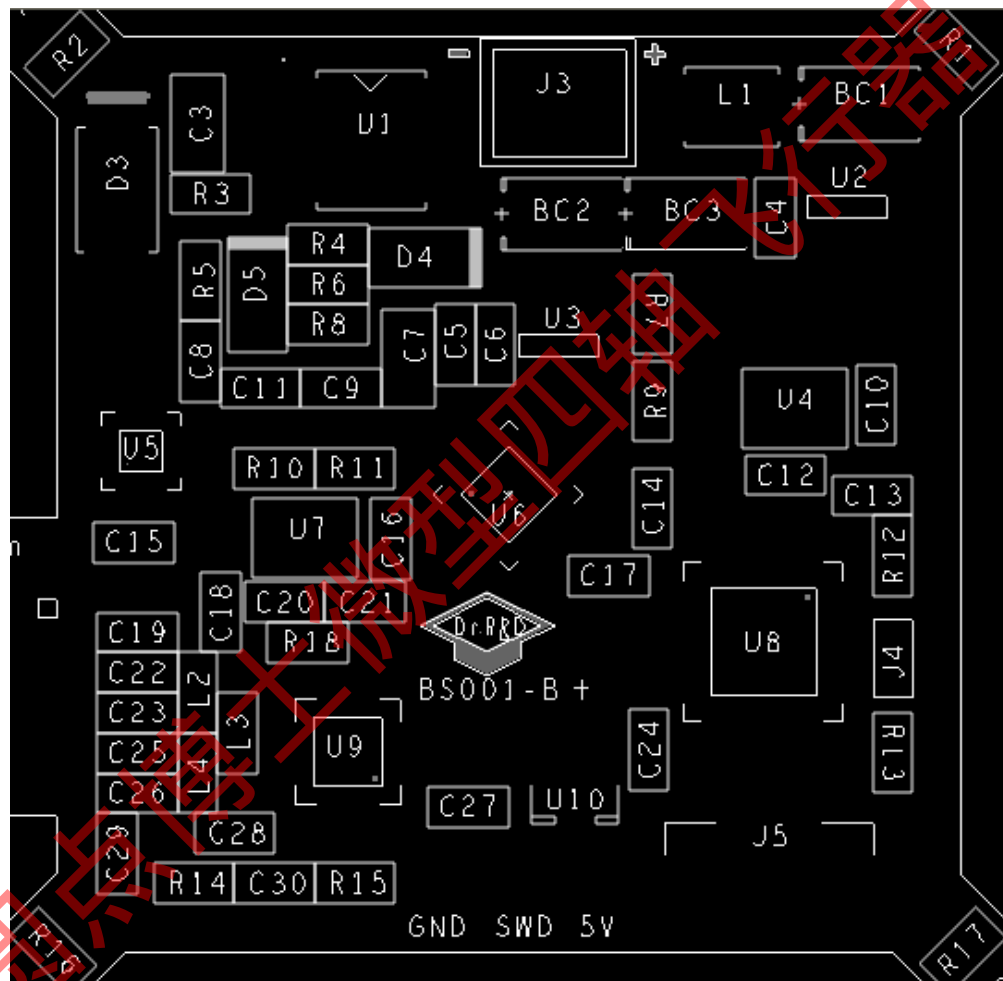


## 圆点博士微型四轴飞行器PCB（六）

DRILL CHART: TOP to BOTTOM			
ALL UNITS ARE IN MILS			
FIGURE	SIZE	PLATED	QTY
+	14.0	PLATED	90
.	14.0	PLATED	1
⊙	32.0	PLATED	6
□	280.0	PLATED	4

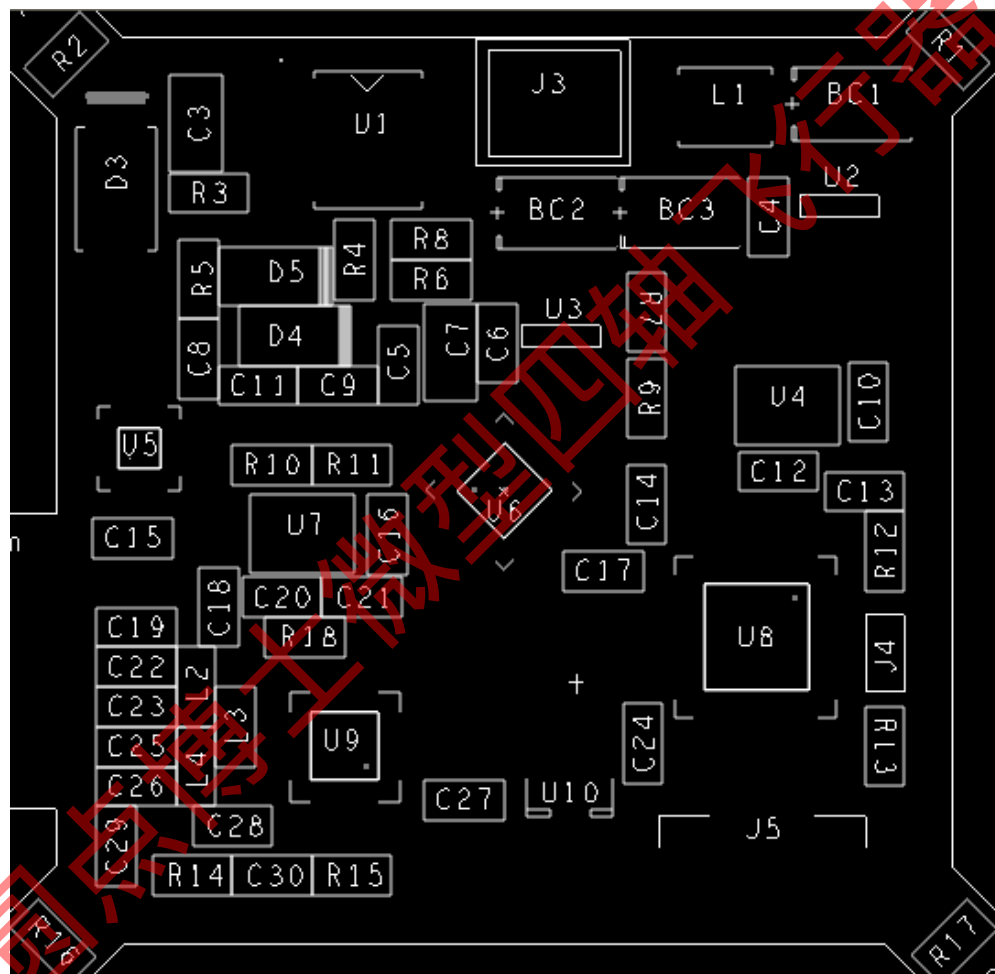
圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器PCB(版本:BS001-B) (七)



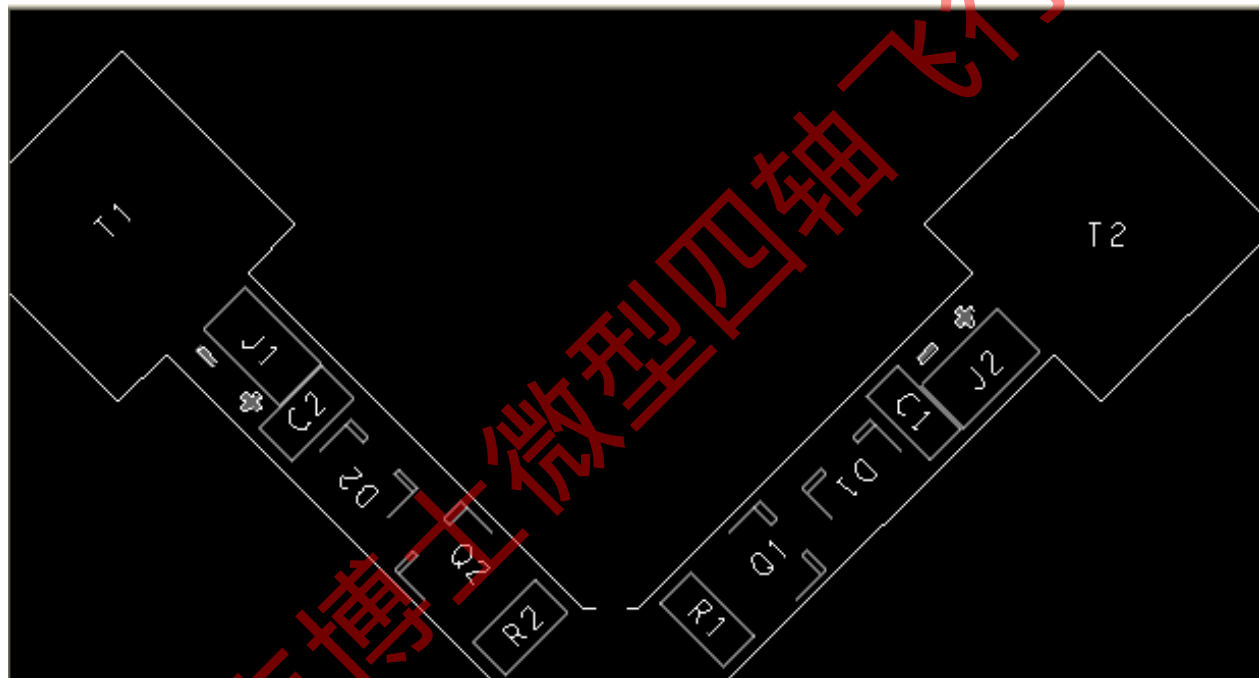
圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器PCB(版本:BS001-C) (八)



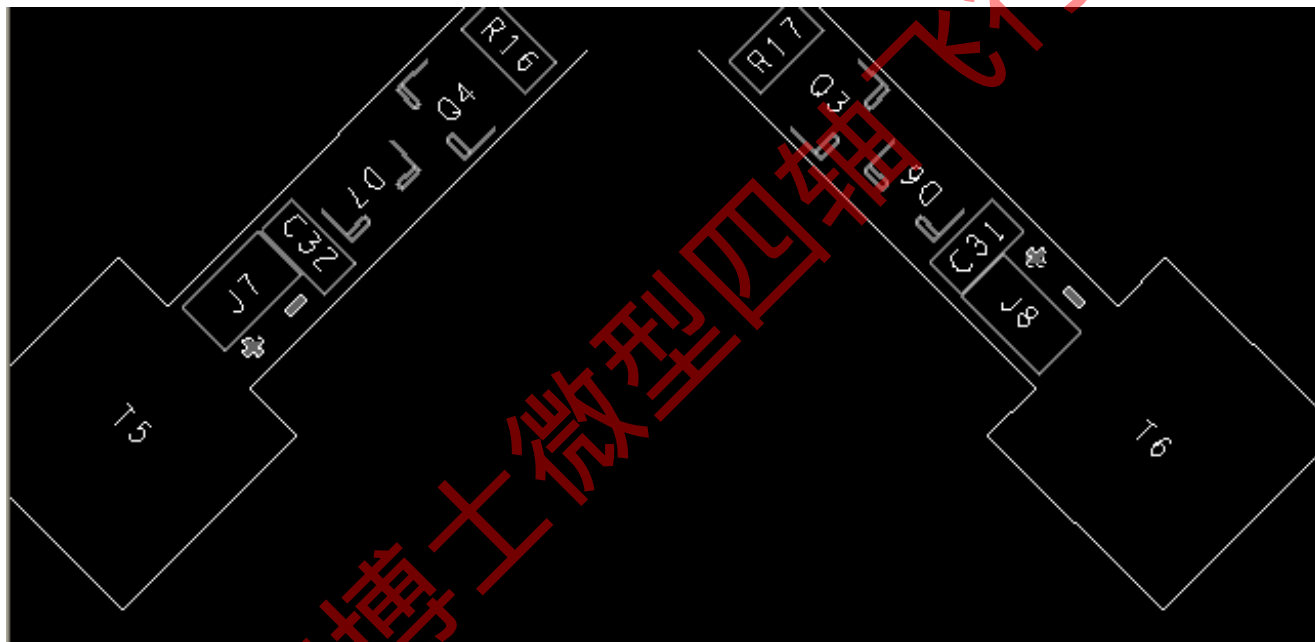
圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器PCB（九）



圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

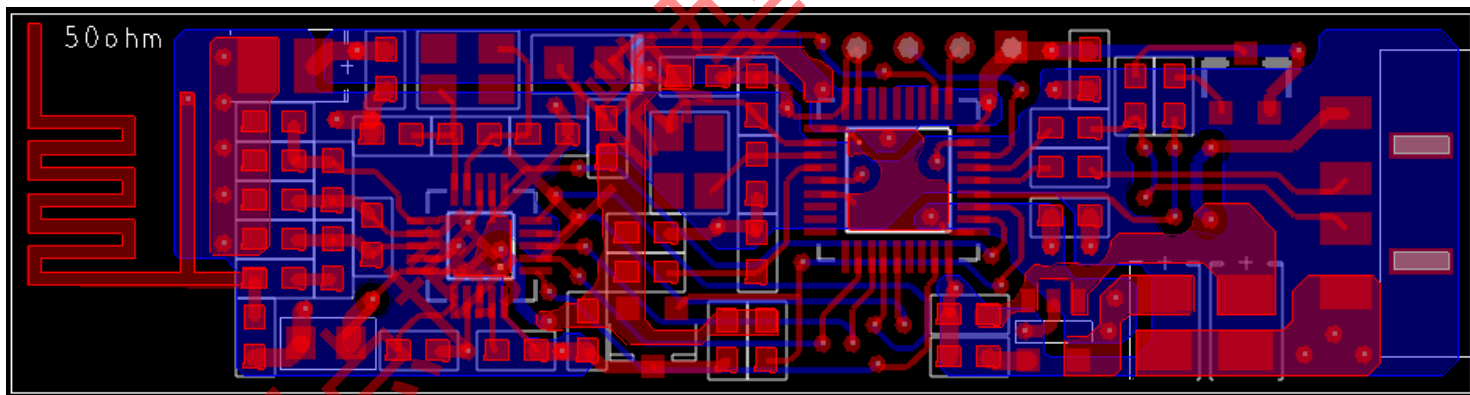
## 圆点博士微型四轴飞行器PCB（十）



圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

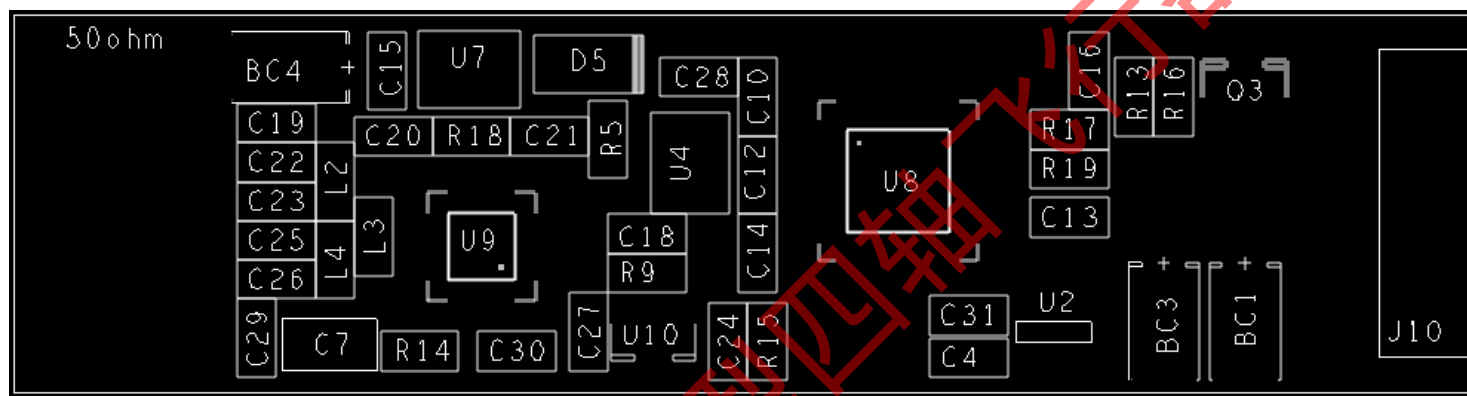
## 圆点博士微型四轴飞行器USB遥控器（十一）

# 圆点博士 微型四轴飞行器



圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器USB遥控器（十二）



圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器USB遥控器元器件（十三）

圆点博士微型四轴飞行器USB遥控器器件清单			
标号	位置	型号	描述
芯片类			
1	U8	STM32F103T8U6	主控芯片
2	U2	SP6205	电压转换LDO
3	U9	NRF24L01+	无线芯片
4	Q3	2N3906	三极管
5	D5	0805 LED	LED指示灯
6	U10	DR2012	软件授权芯片
7	U4,U7	SMD3225 16M	主控芯片使用
8	J10		USB接口

圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>



## 圆点博士微型四轴飞行器USB遥控器元器件（十四）

圆点博士微型四轴飞行器USB遥控器器件清单			
标号	位置	型号	描述
电感			
1	L2	2.7nH	
2	L4	3.9nH	
3	L3	8.2nH	
电阻			
1	R5	470	471
2	R13	1.5K	152
3	R15,R16	2.2K	222
4	R17,R19	22	220
5	R14	22K	223
6	R18	1M	105
7	R9	10K	103

圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器USB遥控器元器件（十五）

圆点博士微型四轴飞行器USB遥控器器件清单			
标号	位置	型号	描述
电容			
1	BC1,BC3,BC4	47uF	
2	C4	0.01uF	103
3	C13,C15,C16,C18,C24, C27,C28,C31	0.1uF	104
4	C14	1uF	105
5	C7	10uF	106
6	C22	2.2nF	222
7	C30	33nF	333
8	C26	1.5pF	
9	C19	4.7pF	
10	C10,C12,C20,C21	22pF	220
11	C23,C25,C29	无（不上）	

圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器专用液晶屏（一）



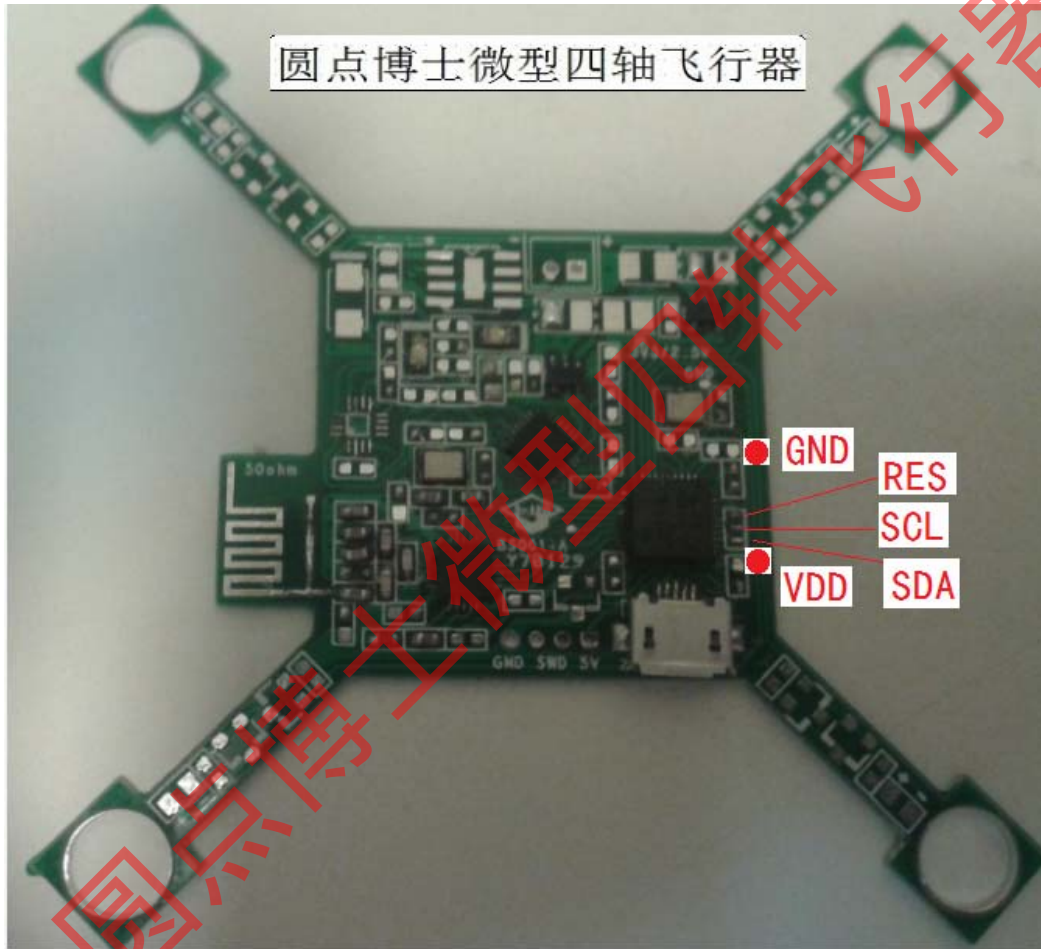
PIN NO.	SYMBOL
1	NC
2	NC
3	SDA
4	VDD
5	VDD
6	SCL
7	VSS
8	NC
9	VSS
10	RES

>>> 仅需连接4根线：

- 1) I2C的SDA （第3脚）
- 2) I2C的SCL （第6脚）
- 3) RESET信号 （第10脚）
- 4) GND信号 （第7,9脚）
- 5) 3.3V电源 （第4,5脚）

>>> 另外需要一个额外的电阻上拉SDA到VDD,阻值范围为1K到10K

## 圆点博士微型四轴飞行器专用液晶屏（二）



圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器专用液晶屏（三）



圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

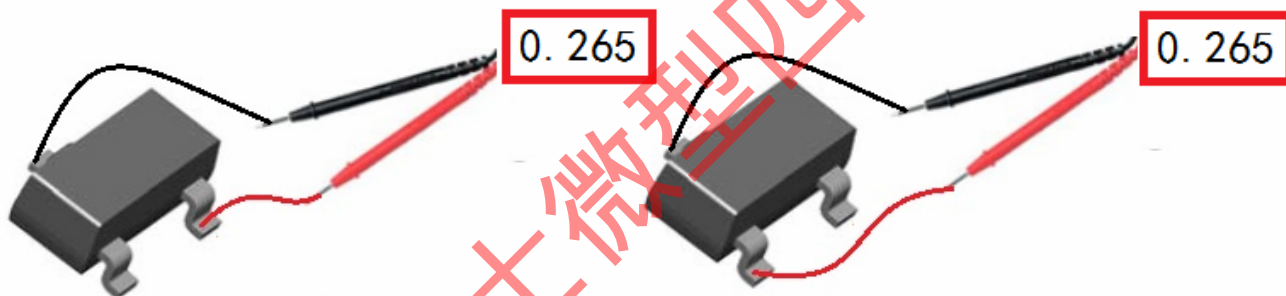
## 圆点博士微型四轴飞行器如何区分二极管(一)

BAT54C属于二极管，用数字万用表测量二极管时，它测量的是二极管的正向电压值

首先把数字万用表调到二极管档位：



然后把万用表按下面示意图进行测量：

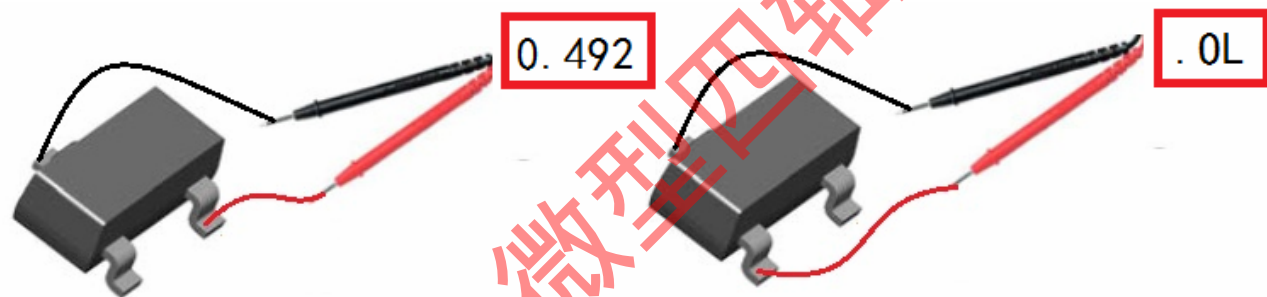


两个读数均不是无穷大(显示1),即表明该芯片是BAT54C，万用表上的读数等于二极管的近似正向压降值，约0.2到0.8.

圆点博士微型四轴飞行器官方网站：<http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器如何区分MOS管(二)

首先把数字万用表调到二极管档位：

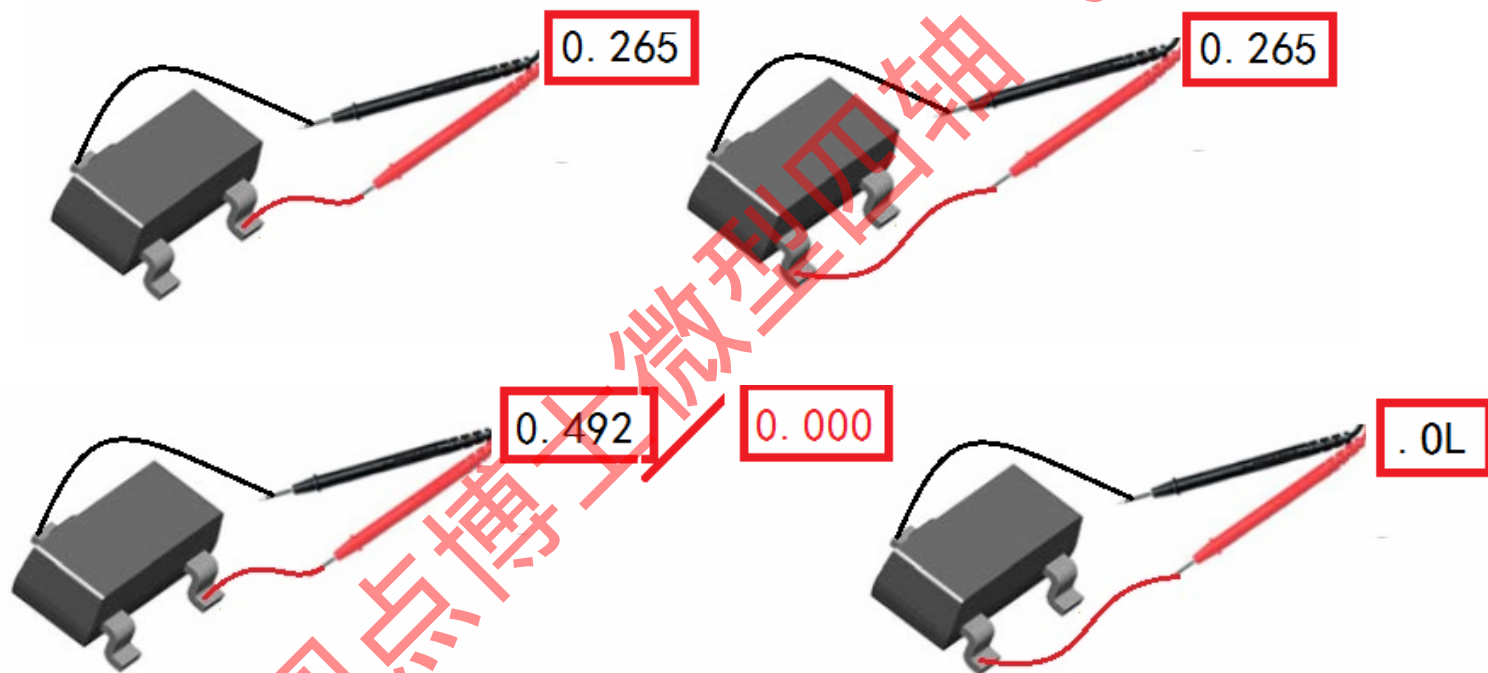


左边图读数不是无穷大(显示1), 右图读数是无穷大(显示1), 即表明该管是SI2302

圆点博士微型四轴飞行器官方网站：<http://www.etootle.com/product/flight-kit.html>

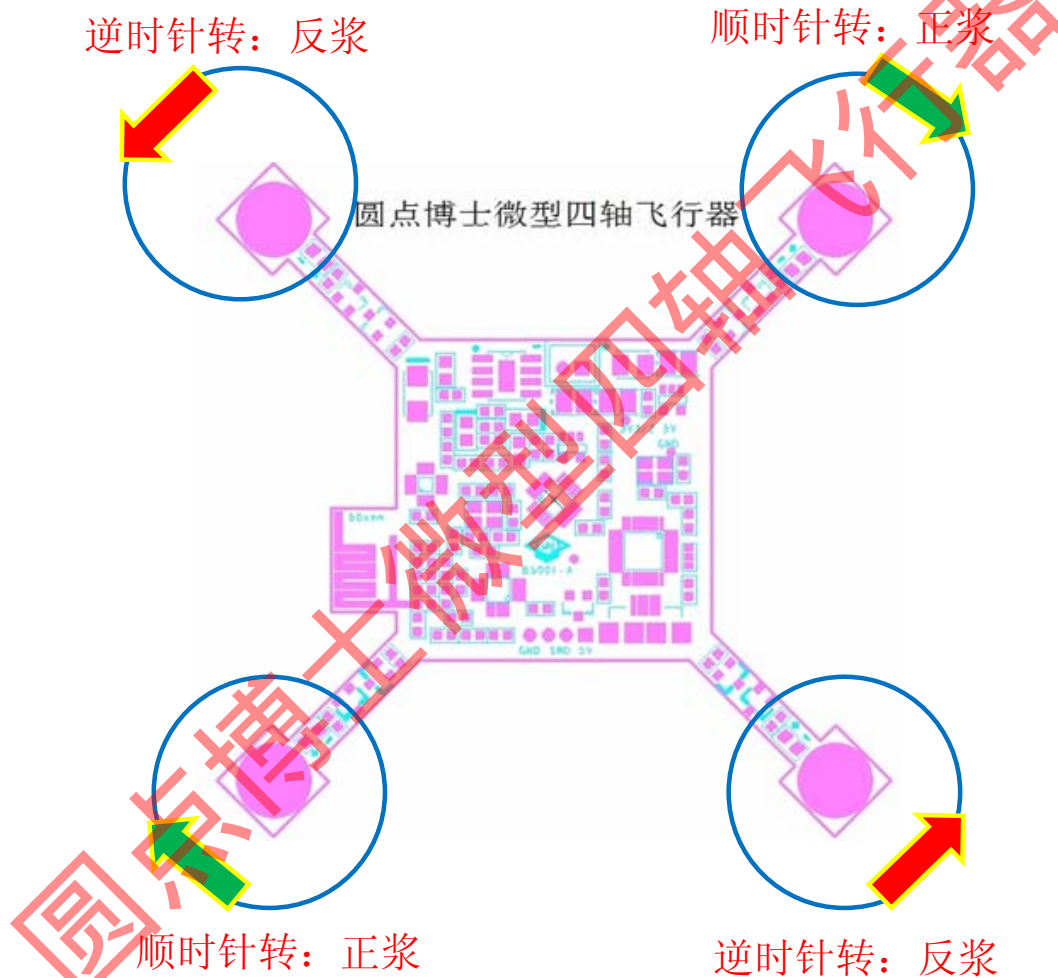
## 圆点博士微型四轴飞行器如何区分MOS管(三)

首先把数字万用表调到二极管档位：



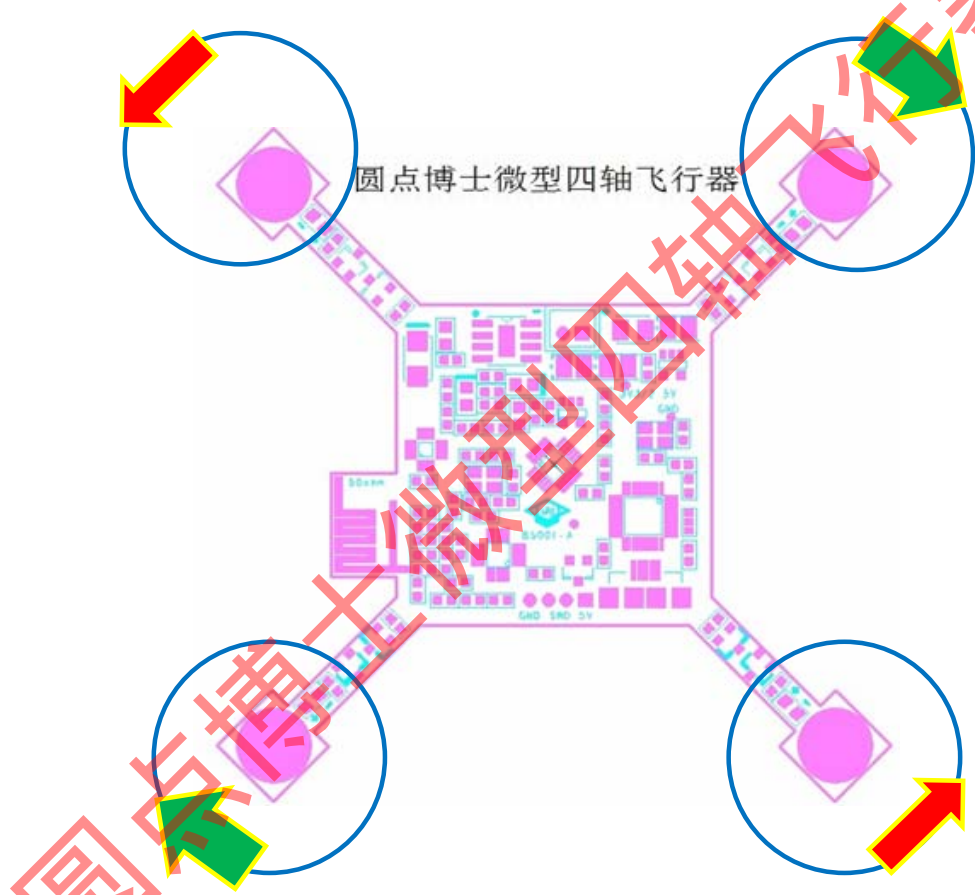


# 圆点博士微型四轴飞行器平衡电机转向图（一）



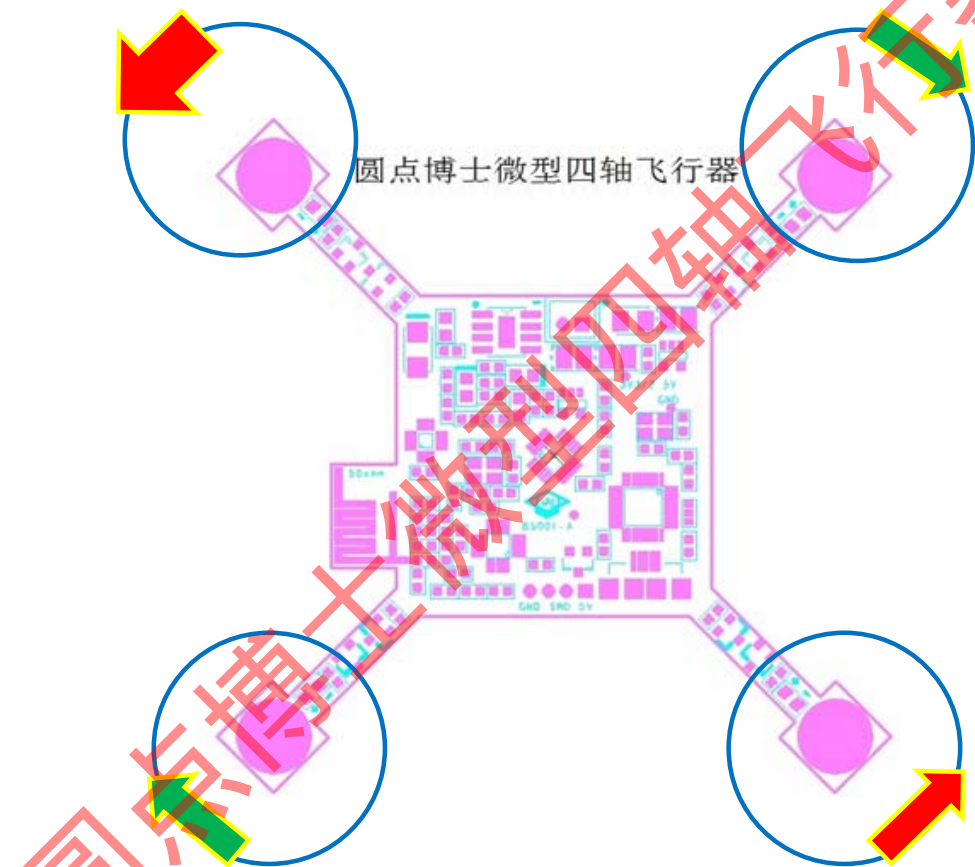
圆点博士微型四轴飞行器官方网站：<http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器旋转电机转向图（二）



圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器前飞电机转向图（三）

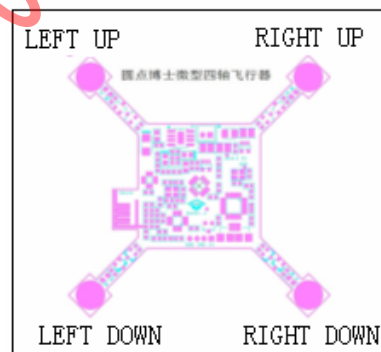


圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

# 圆点博士微型四轴飞行器引脚配置(一)

## 端口A:

#define	PWM_LEFT_DOWN	GPIO_Pin_0
#define	PWM_RIGHT_DOWN	GPIO_Pin_1
#define	PWM_RIGHT_UP	GPIO_Pin_2
#define	PWM_LEFT_UP	GPIO_Pin_3
#define	SPI_WIRELESS_CSN	GPIO_Pin_4
#define	SPI_WIRELESS_SCK	GPIO_Pin_5
#define	SPI_WIRELESS_MISO	GPIO_Pin_6
#define	SPI_WIRELESS_MOSI	GPIO_Pin_7
#define	SPI_WIRELESS_IRQ	GPIO_Pin_8
#define	LED_OUT	GPIO_Pin_15



圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器引脚配置(二)

端口B:

```
#define BAT_Voltage_IN          GPIO_Pin_1
#define SPI_WIRELESS_CE         GPIO_Pin_2

#define I2C_LCD_SDA              GPIO_Pin_3
#define I2C_LCD_SCL              GPIO_Pin_4
#define I2C_LCD_RST              GPIO_Pin_5

#define I2C_SENSOR_SCL           GPIO_Pin_6
#define I2C_SENSOR_SDA           GPIO_Pin_7
```

圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器时钟配置(一)

```
RCC_PLLConfig(RCC_PLLSource_HSE_Div2, RCC_PLLMul_9);
```

注释: 晶体时钟2分频,然后9倍频,相当于 $16\text{MHz}/2*9=72\text{MHz}$

```
RCC_HCLKConfig(RCC_SYSCLK_Div1);
```

```
RCC_PCLK2Config(RCC_HCLK_Div1);
```

```
RCC_PCLK1Config(RCC_HCLK_Div2);
```

```
RCC_ADCCLKConfig(RCC_PCLK2_Div6);
```

```
RCC_PLLCmd(ENABLE);
```

## 圆点博士微型四轴飞行器时钟配置(二)

打开所有IO口相关时钟:

```
RCC_APB2PeriphClockCmd( RCC_APB2Periph_GPIOA |  
                          RCC_APB2Periph_GPIOB |  
                          RCC_APB2Periph_AFIO, ENABLE);
```

打开各模块时钟:

```
RCC_APB2PeriphClockCmd( RCC_APB2Periph_USART1 |  
                          RCC_APB2Periph_SPI1 |  
                          RCC_APB2Periph_ADC1, ENABLE);  
RCC_APB1PeriphClockCmd( RCC_APB1Periph_TIM2 |  
                          RCC_APB1Periph_I2C1, ENABLE);
```

```
TIM_InternalClockConfig(TIM2);
```

关闭JTAG,使用SWD:

```
GPIO_PinRemapConfig(GPIO_Remap_SWJ_JTAGDisable, ENABLE);
```

圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

# 圆点博士微型四轴飞行器点亮LED

设置IO口:

```
GPIO_InitStructure.GPIO_Pin = LED_OUT;  
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;  
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;  
GPIO_Init(GPIOA, &GPIO_InitStructure);
```

循环开关LED灯:

```
while(1)  
{  
    GPIO_ResetBits(GPIOA, LED_OUT);  
    Delay(100000);  
    GPIO_SetBits(GPIOA, LED_OUT);  
    Delay(100000);  
}
```



# 圆点博士微型四轴飞行器点亮LCD液晶屏(一)

## LCD数字显示编码:

```
const u8 number6x8[][6] = {  
    { 0x00, 0x3E, 0x51, 0x49, 0x45, 0x3E }, // 0  
    { 0x00, 0x00, 0x42, 0x7F, 0x40, 0x00 }, // 1  
    { 0x00, 0x42, 0x61, 0x51, 0x49, 0x46 }, // 2  
    { 0x00, 0x21, 0x41, 0x45, 0x4B, 0x31 }, // 3  
    { 0x00, 0x18, 0x14, 0x12, 0x7F, 0x10 }, // 4  
    { 0x00, 0x27, 0x45, 0x45, 0x45, 0x39 }, // 5  
    { 0x00, 0x3C, 0x4A, 0x49, 0x49, 0x30 }, // 6  
    { 0x00, 0x01, 0x71, 0x09, 0x05, 0x03 }, // 7  
    { 0x00, 0x36, 0x49, 0x49, 0x49, 0x36 }, // 8  
    { 0x00, 0x06, 0x49, 0x49, 0x29, 0x1E }, // 9  
    { 0x00, 0x7C, 0x12, 0x11, 0x12, 0x7C }, // A  
    { 0x00, 0x7F, 0x49, 0x49, 0x49, 0x36 }, // B  
    { 0x00, 0x3E, 0x41, 0x41, 0x41, 0x22 }, // C  
    { 0x00, 0x7F, 0x41, 0x41, 0x22, 0x1C }, // D  
    { 0x00, 0x7F, 0x49, 0x49, 0x49, 0x41 }, // E  
    { 0x00, 0x7F, 0x09, 0x09, 0x09, 0x01 }, // F  
};
```

圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器点亮LCD液晶屏(二)

打开**LCD IO**:

```
GPIO_InitStructure.GPIO_Pin = I2C_LCD_RST | I2C_LCD_SCL;  
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;  
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;  
GPIO_Init(GPIOB, &GPIO_InitStructure);
```

```
GPIO_InitStructure.GPIO_Pin = I2C_LCD_SDA;  
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;  
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_OD;  
GPIO_Init(GPIOB, &GPIO_InitStructure);
```

复位**LCD**:

```
GPIO_ResetBits(GPIOB, I2C_LCD_RST);  
Delay(100000);  
GPIO_SetBits(GPIOB, I2C_LCD_RST);
```

## 圆点博士微型四轴飞行器点亮LCD液晶屏(三)

软件模拟I2C ---开始:

```
void I2C_LCD_START(void)
{
    I2C_SDA_1;
    I2C_NOP;
    I2C_SCL_1;
    I2C_NOP;
    I2C_SDA_0;
    I2C_NOP;
    I2C_SCL_0;
    I2C_NOP;
}
```

## 圆点博士微型四轴飞行器点亮LCD液晶屏(四)

软件模拟I2C --- 结束:

```
void I2C_LCD_STOP(void)
{
    I2C_SDA_0;
    I2C_NOP;
    I2C_SCL_1;
    I2C_NOP;
    I2C_SDA_1;
    I2C_NOP;
}
```

## 圆点博士微型四轴飞行器点亮LCD液晶屏(五)

软件模拟I2C --- 发送数据:

```
u8 I2C_LCD_SendByte(u8 Data)
```

```
{  
    u8 i;  
    I2C_SCL_0;  
    for(i=0;i<8;i++)  
    {  
        if(Data&0x80) I2C_SDA_1;  
        else I2C_SDA_0;  
  
        Data<<=1;  
        I2C_NOP;  
  
        I2C_SCL_1;  
        I2C_NOP;  
        I2C_SCL_0;  
        I2C_NOP;  
    }  
}
```

## 圆点博士微型四轴飞行器点亮LCD液晶屏(六)

软件模拟I2C --- 发送数据:

```
I2C_SCL_1;  
I2C_NOP;  
I2C_SCL_0;  
}
```

## 圆点博士微型四轴飞行器点亮LCD液晶屏(七)

初始化LCD:

```
void I2C_LCD_Init(void)
{
    I2C_LCD_START();
    I2C_LCD_SendByte(0x78);
    I2C_LCD_SendByte(0x20);
    I2C_LCD_SendByte(0x2e);
    I2C_LCD_SendByte(0xE8);
    I2C_LCD_SendByte(0x81);
    I2C_LCD_SendByte(0xA0);
    I2C_LCD_SendByte(0xAF);
    I2C_LCD_SendByte(0x40);
    I2C_LCD_STOP();
}
```

## 圆点博士微型四轴飞行器点亮LCD液晶屏(八)

**LCD清屏:**

```
void I2C_LCD_ClrScreen(void)
{
    u16 k;
    I2C_LCD_START();
    I2C_LCD_SendByte(0x7A);
    for(k=0;k<816;k++) I2C_LCD_SendByte(0x00);
    I2C_LCD_STOP();
}
```



## 圆点博士微型四轴飞行器点亮LCD液晶屏(九)

**LCD设置坐标:**

```
void I2C_LCD_SETXY(u8 dir_v,u8 dir_h)
{
    unsigned char tmp,tmp_low,tmp_high;
    tmp=dir_h*12;
    tmp_low=tmp & 0x0f;
    tmp_high=(tmp & 0x70)>>4;
    I2C_LCD_START();
    I2C_LCD_SendByte(0x78);
    I2C_LCD_SendByte(0xb0 | dir_v);
    I2C_LCD_SendByte(0x40);
    I2C_LCD_SendByte(tmp_low);
    I2C_LCD_SendByte(0x10 | tmp_high);
    I2C_LCD_STOP();
}
```

# 圆点博士微型四轴飞行器点亮LCD液晶屏(十)

## LCD显示数据:

```
void I2C_LCD_Show_Num(u8 dis_num)
{
    u8 i,num_low,num_high;

    I2C_LCD_START();
    I2C_LCD_SendByte(0x7A);
    num_high=(dis_num & 0xf0)>>4;
    num_low=dis_num & 0x0f;
    for(i=0;i<6;i++)    I2C_LCD_SendByte(number6x8[num_high][i]);
    for(i=0;i<6;i++)    I2C_LCD_SendByte(number6x8[num_low][i]);
    I2C_LCD_STOP();
}
```

## 圆点博士微型四轴飞行器电池电压换算

```
bat_voltage_value=ADC_GetConversionValue(ADC1);  
voltage_temp=bat_voltage_value;  
voltage_temp=(voltage_temp*3300)>>11;
```

```
voltage_3th=voltage_temp/1000;  
voltage_temp=voltage_temp%1000;
```

```
voltage_2th=voltage_temp/100;  
voltage_temp=voltage_temp%100;  
voltage_1th=voltage_temp/10;  
voltage_0th=voltage_temp%10;
```

```
I2C_LCD_SETXY(0x04,0x02);  
I2C_LCD_Show_Num(voltage_3th);  
I2C_LCD_Show_Char('.');  
I2C_LCD_Show_Num((voltage_2th<<4)|voltage_1th);
```

# 圆点博士微型四轴飞行器MPU6050编程(一)

MPU6050初始化地址:

**#define MPU6050\_ID\_ADDR 0x75**

注: 设备标识I2C地址, 默认数值是0x68, 可用于检测I2C操作是否正常

**#define PWR\_CFG\_ADDR 0x6B**

注: 电源管理I2C地址, 典型值: 0x00(正常启用)

**#define SMPLING\_RATE\_ADDR 0x19**

注: 陀螺仪采样率I2C地址, 典型值: 0x07(125Hz)

**#define FILTER\_CFG\_ADDR 0x1A**

注: 低通滤波频率I2C地址, 典型值: 0x06(5Hz)

**#define ACCEL\_CFG\_ADDR 0x1C**

注: 加速计自检及测量范围I2C地址, 典型值: 0x01(不自检, 2G, 5Hz)

**#define GYRO\_CFG\_ADDR 0x1B**

注: 陀螺仪自检及测量范围I2C地址, 典型值: 0x18(不自检, 2000deg/s)

圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器MPU6050编程(二)

加速度X,Y,Z轴，数据长度是2个字节

```
#define ACCEL_XOUT_H 0x3B
#define ACCEL_XOUT_L 0x3C
#define ACCEL_YOUT_H 0x3D
#define ACCEL_YOUT_L 0x3E
#define ACCEL_ZOUT_H 0x3F
#define ACCEL_ZOUT_L 0x40
```

陀螺仪X,Y,Z轴，数据长度是2个字节

```
#define GYRO_XOUT_H 0x43
#define GYRO_XOUT_L 0x44
#define GYRO_YOUT_H 0x45
#define GYRO_YOUT_L 0x46
#define GYRO_ZOUT_H 0x47
#define GYRO_ZOUT_L 0x48
```

温度值，数据长度是2个字节

```
#define TEMP_OUT_H 0x41
#define TEMP_OUT_L 0x42
```

## 圆点博士微型四轴飞行器MPU6050编程(三)

MPU6050 I2C 设备是：1101 000X (二进制)，具体到16进制，  
写地址是0xD0 (MPU6050\_DEV\_WR\_ADDR=0xD0)

MPU6050 I2C写数据：

```
void MPU6050_I2C_WR (u8 MPU6050_REG_Addrs,u8 MPU6050_REG_data)
{
    I2C_Start();
    I2C_SendByte(MPU6050_DEV_WR_ADDR);
    I2C_SendByte(MPU6050_REG_Address);
    I2C_SendByte(MPU6050_REG_data);
    I2C_Stop();
}
```

圆点博士微型四轴飞行器官方网站：<http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器MPU6050编程(四)

MPU6050 I2C 设备是：1101 000X (二进制)，具体到16进制，读地址是0xD1 (MPU6050\_DEV\_RD\_ADDR=0xD1)

MPU6050 I2C读数据：

u8 MPU6050\_I2C\_RD (u8 MPU6050\_REG\_Addrs)

```
{  
    I2C_Start();  
    I2C_SendByte(MPU6050_DEV_WR_ADDR);  
    I2C_SendByte(MPU6050_REG_Address);  
    I2C_Start();  
    I2C_SendByte(MPU6050_DEV_RD_ADDR);  
    MPU6050_REG_data=I2C_ReciveByte();  
    I2C_Stop();  
    return MPU6050_REG_data;  
}
```

圆点博士微型四轴飞行器官方网站：<http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器MPU6050编程(五)

初始化MPU6050:

```
void MPU6050_Init ()
{
    MPU6050_I2C_WR(PWR_CFG_ADDR, 0x00);
    MPU6050_I2C_WR (SMPLING_RATE_ADDR, 0x07);
    MPU6050_I2C_WR (FILTER_CFG_ADDR, 0x06);
    MPU6050_I2C_WR (ACCEL_CFG_ADDR, 0x01);
    MPU6050_I2C_WR (GYRO_CFG_ADDR, 0x18);
}
```

读取MPU6050数据:

```
u16 MPU6050_Get_Data(u8 MPU6050_REG_Address)
{
    u8 DATA_H, DATA_L;
    DATA_H = MPU6050_I2C_RD(MPU6050_REG_Address);
    DATA_L = MPU6050_I2C_RD(MPU6050_REG_Address+1);
    return (DATA_H << 8) + DATA_L;
}
```

圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>



## 圆点博士微型四轴飞行器遥控器编程(一)

```
void SPI_Init(void)
{
    SPI_InitTypeDef      SPI_InitStructure;
    SPI_I2S_DeInit(SPI1);
    SPI_Cmd(SPI1, DISABLE);
    SPI_InitStructure.SPI_Direction = SPI_Direction_2Lines_FullDuplex;
    SPI_InitStructure.SPI_Mode = SPI_Mode_Master;
    SPI_InitStructure.SPI_DataSize = SPI_DataSize_8b;
    SPI_InitStructure.SPI_CPOL = SPI_CPOL_Low;
    SPI_InitStructure.SPI_CPHA = SPI_CPHA_1Edge;
    SPI_InitStructure.SPI_NSS = SPI_NSS_Soft;
    SPI_InitStructure.SPI_BaudRatePrescaler = SPI_BaudRatePrescaler_4;
    SPI_InitStructure.SPI_FirstBit = SPI_FirstBit_MSB;
    SPI_InitStructure.SPI_CRCPolynomial = 7;

    SPI_Init(SPI1, &SPI_InitStructure);
    SPI_Cmd(SPI1, ENABLE);
}
```

圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器遥控器编程(二)

使用NRF默认地址: 0xE7 0xE7 0xE7 0xE7 0xE7

```
void RX_Mode(void)
{
    CE_L();
    SPI_WR_Reg(WRITE_REG + EN_AA, 0x01); // 0x01; // 频道0自动 ACK应答允许
    SPI_WR_Reg(WRITE_REG + EN_RXADDR, 0x01); // 0x01; // 允许接收地址只有频道0.
    SPI_WR_Reg(WRITE_REG + SETUP_RETR, 0x11); // 0x11; // 设置自动重发时间和次数.
    SPI_WR_Reg(WRITE_REG + RF_CH, 40); // 40; // 设置信道工作为2.4GHZ.
    SPI_WR_Reg(WRITE_REG + RX_PW_P0, 32); // 32; // 设置接收数据长度.
    SPI_WR_Reg(WRITE_REG + RF_SETUP, 0x0F); // 0x0F; // 设置发射速率为

    SPI_WR_Reg(WRITE_REG + CONFIG, 0x0F);
    CE_H();
}
```

圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器遥控器编程(三)

使用NRF默认地址: 0xE7 0xE7 0xE7 0xE7 0xE7

```
void TX_Mode(void)
```

```
{  
    CE_L();  
    SPI_Write_Buf(WRITE_REG + TX_ADDR, TX_ADDRESS, TX_ADR_WIDTH);  
    SPI_Write_Buf(WRITE_REG + RX_ADDR_P0, RX_ADDRESS, RX_ADR_WIDTH);  
  
    SPI_WR_Reg(WRITE_REG + EN_AA, 0x01); // 频道0自动 ACK应答允许  
    SPI_WR_Reg(WRITE_REG + EN_RXADDR, 0x01); // 允许接收地址只有频道0.  
    SPI_WR_Reg(WRITE_REG + SETUP_RETR, 0x11); // 设置自动重发时间和次数.  
    SPI_WR_Reg(WRITE_REG + RF_CH, 40); // 设置信道工作为2.4GHZ  
    SPI_WR_Reg(WRITE_REG + RF_SETUP, 0x0F); // 设置发射速率  
    SPI_WR_Reg(WRITE_REG + CONFIG, 0x0E);  
    CE_H();  
}
```

圆点博士微型四轴飞行器官方网站: <http://www.etootle.com/product/flight-kit.html>

## 圆点博士微型四轴飞行器遥控器编程(四)

```
if(nRF24L01_RxPacket(RxBuf))    //NRF接收到数据
{
    if((RxBuf[0]==0x5A)&&(RxBuf[1]==0xA5)) //数据头检查
    {
        for(i=0;i<32;i++) my_receive_buffer[i]=RxBuf[i]; //拷贝数据
        NRF_MsgCmd_H=RxBuf[2]; //拷贝命令字
        NRF_MsgCmd_L=RxBuf[3]; //拷贝命令字

        NRF_Data_Send_Req=1;    //设置发送数据标志位
        NRF24_TX_Mode();        //进入数据发送模式
    }
}
```

## 圆点博士微型四轴飞行器遥控器编程(五)

```
Get_Flight_Parameters(MPU6050_data_buffer);    //读传感器数据  
for(i=0;i<28;i++) TxBuf[i]=MPU6050_data_buffer[i]; //把数据拷贝到无线发送缓冲
```

```
if(NRF_Data_Send_Req==1)  
{  
    NRF_Data_Send_Req=0;    //清无线发送请求标志  
    NRF24L01_TxPacket(TxBuf); //发送数据  
    Delay(1000);  
  
    NRF24_RX_Mode();    //进入接收模式  
    Delay(1000);  
}
```

## 圆点博士微型四轴飞行器遥控器编程(六)

```
if((NRF_MsgCmd_H==0x8A)&&(NRF_MsgCmd_L==0x75))
{
    NRF_MsgCmd_H=0;
    NRF_MsgCmd_L=0;

    ...打开电机...
}
else if ((NRF_MsgCmd_H==0x7C)&&(NRF_MsgCmd_L==0x83))
{
    NRF_MsgCmd_H=0;
    NRF_MsgCmd_L=0;

    ...关闭电机...

}
```