PART A

功能实现：将以下程序编译、下载，打开串口监视器，波特率设置为9600，波特率左边要选择换行符或者回车；按下复位键，在输入窗口写入任意字符串，点击发送，即可看到打印出输入的内容。

String inString = "";

void setup() {

Serial.begin(9600);

}

void loop() {

while (Serial.available() > 0) {

int inChar = Serial.read();

inString += (char)inChar;

if (inChar=='\n' || inChar=='\r') {

Serial.print(inString);

inString = "";

}

}

delay(10);

}

程序分析：字符串inString用来存储串口收到的字符，Serial.begin(9600)中的9600要和串口监视器中的波特率保持一致；当有字符输入时，Serial.available()会大于零，进入到while循环中，用Serial.read()读取到字符，并将其赋值给inChar，然后添加到字符串inString中；上文绿色部分（串口监视器）的作用为：给输入的字符串末尾添加换行符或者回车符，换行符为’\n’，回车为’\r’，所以通过判断字符inChar是否为’\n’或者’\r’来判断输入是否结束，结束之后即可输出字符串inString，同时将inString幅值为空字符串。

PART B

硬件连接：将RGB灯的RGB三个引脚分别接到板子上的13、12、14引脚上，具体13、12、14引脚对应到板子的哪个引脚可参照图片“nodemcu引脚图.PNG”，将RGB灯的GND接到板子的GND即可。

功能实现：将以下程序编译下载，上电之后，即可看到RGB灯会亮红色一秒、绿色一秒、蓝色一秒，依次循环。PART B部分功能验证完毕。

const int PIN\_LED\_RED = 13;

const int PIN\_LED\_GREEN = 12;

const int PIN\_LED\_BLUE = 14;

void setup() {

Serial.begin(9600);

pinMode(PIN\_LED\_RED, OUTPUT);

pinMode(PIN\_LED\_GREEN, OUTPUT);

pinMode(PIN\_LED\_BLUE, OUTPUT);

}

void loop() {

analogWrite(PIN\_LED\_RED, 255\*255\*0.041);

analogWrite(PIN\_LED\_GREEN, 0);

analogWrite(PIN\_LED\_BLUE, 0);

delay(1000);

analogWrite(PIN\_LED\_RED, 0);

analogWrite(PIN\_LED\_GREEN, 255\*255\*0.041);

analogWrite(PIN\_LED\_BLUE, 0);

delay(1000);

analogWrite(PIN\_LED\_RED, 0);

analogWrite(PIN\_LED\_GREEN, 0);

analogWrite(PIN\_LED\_BLUE, 255\*255\*0.041);

delay(1000);

}

程序讲解：利用函数pinMode(PIN\_LED\_RED, OUTPUT)将三个引脚设置为输出；利用函数analogWrite()将R、G、B三个颜色值和亮度值写入到三个引脚上，可参考公式analogWrite(PIN\_LED\_RED, red\*bright\*0.041)，之后RGB灯即可显示对应的颜色。以上示例将RGB灯设置为红色一秒、绿色一秒、蓝色一秒循环播放。

PART C

功能实现：将以下程序编译、下载，打开串口监视器，按下复位键，按照log提示依次输入R、G、B和亮度值，即可看到RGB灯按照输入的值发出对应的光，同时可循环输入R、G、B值，再次让小灯亮。

String inString = "";

int n = 0;

int red = 0;

int green = 0;

int blue = 0;

int bright = 0;

const int PIN\_LED\_RED = 13;

const int PIN\_LED\_GREEN = 12;

const int PIN\_LED\_BLUE = 14;

void setup() {

Serial.begin(9600);

pinMode(PIN\_LED\_RED, OUTPUT);

pinMode(PIN\_LED\_GREEN, OUTPUT);

pinMode(PIN\_LED\_BLUE, OUTPUT);

Serial.println();

Serial.print("input the RED value:");

}

void loop() {

while (Serial.available() > 0) {

int inChar = Serial.read();

inString += (char)inChar;

if (inChar=='\n' || inChar=='\r') {

if(n == 0){

red = inString.toInt();

Serial.println(red);

Serial.print("input the GREEN value:");

}

else if(n == 1){

green = inString.toInt();

Serial.println(green);

Serial.print("input the BLUE value:");

}

else if(n == 2){

blue = inString.toInt();

Serial.println(blue);

Serial.print("input the BRIGHTNESS value:");

}

else{

bright = inString.toInt();

Serial.println(bright);

analogWrite(PIN\_LED\_RED, red\*bright\*0.041);

analogWrite(PIN\_LED\_GREEN, green\*bright\*0.041);

analogWrite(PIN\_LED\_BLUE, blue\*bright\*0.041);

Serial.println("the led is already OPEN!");

Serial.println("you can input the RED GREEN BLUE and BRIGHTNESS once again!");

Serial.print("input the RED value:");

n = -1;

}

n++;

inString = "";

}

}

delay(10);

}

程序讲解：该部分程序为PART A与PART B相结合，再加入一些控制逻辑而实现的效果。变量red、green、blue、bright分别用来存储小灯的红、绿、蓝和亮度值；变量n用来区别输入的R、G、B值和亮度值，其初始值为0，对应R值，然后为1，对应G值，以此类推，依次输入各个值；当所有值到得到之后，即可调用analogWrite()将颜色值和亮度值输出到对应的引脚上，小灯即可发出特定的光；然后输出log，提示下次的输入，并将变量n初始化，准备下一次输入；bright=inString.toInt()的作用为，将字符串inString转换成int类型的变量，并将结果赋值给bright。