Blinker APP:

添加设备：进入app主界面，点击右上角加号-->独立设备-->wifi接入-->阿里云-->记录下Secret Key-->返回我的设备

(尽量保持 APP主界面中“所有设备“下只有一个设备)

米家 APP:

我的-->其他平台设备-->点击添加（若已经绑定，则直接同步设备即可）-->找到“点灯科技“-->绑定账号-->输入 Blinker 账号密码，点击绑定-->点击同步设备（显示”同步成功“，并且所显示设备的名称和Blinker APP上的设备名称一致，说明设备绑定成功）

小爱同学 APP:

利用米家账号登陆即可

PART A：

功能实现：将以下程序编译下载，打开串口监视器。

安卓手机：打开小爱同学APP，点击右上角个人中心，在小爱设置中点击语音唤醒，将语音唤醒打开（根据需要选择APP内唤醒、后台唤醒）；返回，点击连续对话设置，开启连续对话模式（这一步也根据需要自行选择，建议打开）；然后说“小爱同学”，小爱有回应之后，并且显示“倾听中”，说“把灯调为红色（绿色、蓝色）”，在串口监视器中可以看到输出颜色所对应的RGB值（蓝色—> colorR:0, colorG:0, colorB:255)；同样的方法说出“把灯的亮度调成百分之八十”，在串口监视器中可以看到输出“colorW:80”；至此，Android机PART A部分验证完毕。

苹果手机：打开小爱同学APP，点击右上角个人中心，在小爱设置中点击“连续对话设置”，开启“连续对话模式”（这一步根据需要自行选择，建议打开）；返回到主页之后，看到中间下方，有一个彩色圆圈一直在转动，点击它，看到显示“倾听中”时，说出“把灯调为红色（绿色、蓝色）”，在串口监视器中可以看到输出颜色所对应的RGB值（蓝色—> colorR:0, colorG:0, colorB:255)；同样的方法说出“把灯的亮度调成百分之八十”，在串口监视器中可以看到输出“colorW:80”；至此，苹果手机PART A部分验证完毕。

#define BLINKER\_WIFI

#define BLINKER\_MIOT\_LIGHT

#include <Blinker.h>

const char auth[] = "your own Secret Key";

const char ssid[] = "your wifi name ";

const char pswd[] = "your wifi password";

int colorR = 0;

int colorG = 0;

int colorB = 0;

int colorW = 0;

void miotColor(int32\_t color)

{

colorR = color >> 16 & 0xFF;

colorG = color >> 8 & 0xFF;

colorB = color & 0xFF;

Serial.print("colorR:");

Serial.print(colorR);

Serial.print(", colorG:");

Serial.print(colorG);

Serial.print(", colorB:");

Serial.println(colorB);

}

void miotBright(const String & bright)

{

colorW = bright.toInt();

Serial.print("colorW:");

Serial.println(colorW);

}

void setup() {

Serial.begin(115200);

BLINKER\_DEBUG.stream(Serial);

Blinker.begin(auth,ssid,pswd);

BlinkerMIOT.attachColor(miotColor);

BlinkerMIOT.attachBrightness(miotBright);

}

void loop() {

Blinker.run();

}

程序分析：变量colorR、colorG、colorB、colorW分别用来存储设备端收到的R、G、B和亮度值；miotColor()为用来接收灯的颜色的回调函数，当用小爱同学控制灯的颜色的时候，就会进入到这个函数，并将说出的颜色值通过color变量传入到该函数中，然后从变量color中分离出R、G、B值，并通过串口监视器把收到的R、G、B值打印出来；miotBright()为用来接受灯的亮度的回调函数，当用小爱同学控制灯的亮度的时候，就会进入到该函数中，并将灯的亮度值传入到该函数中，并通过串口监视器打印其亮度值（该亮度值的范围为0到100）；BlinkerMIOT.attachColor(miotColor)和BlinkerMIOT.attachBrightness(miotBright)用来注册以上两个回调函数miotColor()和miotBright()，使得在用小爱同学控制设备时能够进入到对应的回调函数中，来输出信息。

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。

功能实现：将以下程序编译下载，打开串口监视器，用PART A的方法来控制灯的颜色和亮度，在串口监视器中能够看到输出对应的颜色和亮度值，同时rgb灯的颜色和亮度也跟着变化，至此实验验证成功。

#define BLINKER\_WIFI

#define BLINKER\_MIOT\_LIGHT

#include <Blinker.h>

const char auth[] = "your own Secret Key";

const char ssid[] = "your wifi name ";

const char pswd[] = "your wifi password";

int colorR = 255;

int colorG = 255;

int colorB = 255;

int colorW = 100;

const int PIN\_LED\_RED = 13;

const int PIN\_LED\_GREEN = 12;

const int PIN\_LED\_BLUE = 14;

void myRGB(int R,int G,int B,int W)

{

int W1;

W1 = map(W,0,100,0,255);

Serial.print("RED:");

Serial.print(R);

Serial.print(", GREEN:");

Serial.print(G);

Serial.print(", BLUE:");

Serial.print(B);

Serial.print(", BRIGHTNESS:");

Serial.println(W1);

analogWrite(PIN\_LED\_RED, R\*W1\*0.041);

analogWrite(PIN\_LED\_GREEN, G\*W1\*0.041);

analogWrite(PIN\_LED\_BLUE, B\*W1\*0.041);

}

void miotColor(int32\_t color)

{

colorR = color >> 16 & 0xFF;

colorG = color >> 8 & 0xFF;

colorB = color & 0xFF;

BlinkerMIOT.color(color);

BlinkerMIOT.print();

myRGB(colorR,colorG,colorB,colorW);

}

void miotBright(const String & bright)

{

colorW = bright.toInt();

BlinkerMIOT.brightness(colorW);

BlinkerMIOT.print();

myRGB(colorR,colorG,colorB,colorW);

}

void setup() {

Serial.begin(115200);

BLINKER\_DEBUG.stream(Serial);

pinMode(PIN\_LED\_RED, OUTPUT);

pinMode(PIN\_LED\_GREEN, OUTPUT);

pinMode(PIN\_LED\_BLUE, OUTPUT);

Blinker.begin(auth,ssid,pswd);

BlinkerMIOT.attachColor(miotColor);

BlinkerMIOT.attachBrightness(miotBright);

myRGB(colorR,colorG,colorB,colorW);

}

void loop() {

Blinker.run();

}

程序分析：变量PIN\_LED\_RED、PIN\_LED\_GREEN、PIN\_LED\_BLUE用来表示小灯的R、G、B三个引脚分别对应到板子上的哪个引脚，这三个值可以自行更改（改完之后要硬件接法也要做对应的更改）；函数void myRGB(int R,int G,int B,int W)的功能为：根据输入的R、G、B值和亮度值，来控制小灯输出对应的颜色和亮度，并将这些值通过串口监视器打印出来；myRGB()函数里边有一个map()函数，其功能为：输入的亮度值W的取值范围为0到100，将其映射到范围为0到255的W1中，因为后边控制小灯的函数analogWrite(PIN\_LED\_RED, R\*W1\*0.041)中的变量W1的范围为0到255；在setup()中初始化R、G、B三个引脚为输出，并调用myRGB()函数将小灯初始化为白色最亮；函数BlinkerMIOT.attachColor(miotColor)功能为注册回调函数miotColor()，当有颜色类控制指令下发时，会进入到该回调函数中；函数BlinkerMIOT.attachBrightness(miotBright)功能为注册回调函数miotBright()，当有亮度控制指令下发时，会进入到该回调函数中；同时在回调函数miotColor()和miotBright()中加入myRGB()函数，当有新的控制数据进来时，能够用myRGB()函数将最新的数据输出到引脚上来控制小灯的亮度和颜色。