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（可以尝试其他选项）-->记录下Secret Key-->返回我的设备

配置设备：进入app主界面，点击上边新建的设备-->点击右上角的铅笔图标

1.点击下边的“按键”（屏幕中会多出一个方框）-->点击方框-->在“数据键名”里边填“btn-on”-->在”按键类型”中选择“开关按键”-->点击右上角“保存”

2.点击下边的“按键”（屏幕中会多出一个方框）-->点击方框-->在“数据键名”里边填“btn-off”-->在”按键类型”中选择“开关按键”-->点击右上角“保存”

3.点击右上角小锁形状的按钮。

硬件连接：

该部分可参考实验一。

PART A：

功能实现：将以下代码编译运行（将之前记录下的Secret Key、即将要连接的WiFi名称和秘密写到对应的位置），打开串口监视器，按下复位键，当看到“MQTT Connected!”，说明板子联网成功。在Blinker APP上按下刚才新建的“btn-on”按键，在串口监视器可以看到“you enter the button 'on'!”，按下”btn-off“按键，在串口监视器可以看到“you enter the button 'off'!”，PART A部分功能验证成功。

#define BLINKER\_WIFI

#include <Blinker.h>

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

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

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

BlinkerButton Button1("btn-on");

BlinkerButton Button2("btn-off");

void button1\_callback(const String & state)

{

Serial.println("you enter the button 'on'!");

}

void button2\_callback(const String & state)

{

Serial.println("you enter the button 'off'!");

}

void setup()

{

Serial.begin(115200);

BLINKER\_DEBUG.stream(Serial);

Blinker.begin(auth, ssid, pswd);

Button1.attach(button1\_callback);

Button2.attach(button2\_callback);

}

void loop() {

Blinker.run();

}

程序分析：BlinkerButton Button1("btn-on")为新建组件Button1，用来开灯；Button1.attach(button1\_callback)为Button1注册回调函数button1\_callback()，当”btn-on”这个按键被按下时，就会进入到这个回调函数button1\_callback()中，从而打印出提示信息；同样的方法也新建组件Button2、并注册回调函数，回调函数内打印信息，用来关灯。

PART B:

功能实现：将程序编译、下载后，按下复位键，看到LED灯按照程序中是时间间隔在交替闪烁，PART B部分功能验证成功。

const int myLed = LED\_BUILTIN;

void setup() {

pinMode(myLed, OUTPUT);

}

void loop() {

digitalWrite(myLed, LOW);

delay(1000);

digitalWrite(myLed, HIGH);

delay(2000);

}

程序分析：pinMode(myLed, OUTPUT)将led对应的引脚设置为输出模式，利用函数digitalWrite(myLed, LOW)将该引脚拉高、拉低，从而控制led的亮或者灭。函数delay(1000)用来延时，单位为毫秒。

PART C:

功能实现：将以下程序编译、下载，按下复位键，打开串口监视器，当显示”MQTT connect”时，说明板子配网成功；按下Blinker APP上是”btn-on”按键，LED变亮，按下”btn-off”键，LED熄灭；当LED处于亮的状态时，再次按下”btn-on”键时，就会输出提示信息”the led is already on…”。实验二功能验证成功。

#define BLINKER\_WIFI

#include <Blinker.h>

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

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

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

BlinkerButton Button1("btn-on");

BlinkerButton Button2("btn-off");

const int myLed = LED\_BUILTIN;

void button1\_callback(const String & state)

{

if(!digitalRead(myLed)){

Serial.println("the led is already on,but you enter the button 'on' second time!\n you can enter the another 'off' button!");

}

else{

Serial.println("you enter the button 'on'!");

digitalWrite(myLed, LOW);

}

}

void button2\_callback(const String & state)

{

if(digitalRead(myLed)){

Serial.println("the led is already off,but you enter the button 'off' second time!\n you can enter the another 'on' button!");

}

else{

Serial.println("you enter the button off!");

digitalWrite(myLed, HIGH);

}

}

void setup()

{

Serial.begin(115200);

BLINKER\_DEBUG.stream(Serial);

pinMode(myLed, OUTPUT);

digitalWrite(myLed, HIGH);

Blinker.begin(auth, ssid, pswd);

Button1.attach(button1\_callback);

Button2.attach(button2\_callback);

}

void loop() {

Blinker.run();

}

程序分析：PART C为PART A与PART B的结合；在回调函数button1\_callback()中采用digitalRead(myLed)函数来判断LED的状态，button1\_callback()为”btn-on”的回调函数，当按下”btn-on”后，先判断LED的状态，如果灯已经开了，则输出提示信息”the led is already on…”；否则，利用函数digitalWrite(myLed, LOW)来将LED的引脚设置为低电平，从而将灯打开；同样的方法在关灯的回调函数button2\_callback()也实现上述同样的功能。