

WiFi 模块

BMC81M001 Arduino Library V1.0.3 说明

版本: V1.20 日期: 2023-12-20

www.bestmodulescorp.com



目录

简介	3
Arduino Lib 函数	3
Arduino Lib 下载及安裝	8
Arduino 范例	9
范例 1: TCP	9
范例 2:Alinyun_Iot	11
范例 3: ThingSpeak	
范例 4: ThingSpeakPublish	
范例 5: ThingSpeakSubscribe	



🌓 简介

BMC81M001 是倍创推出的 WiFi 模块,使用 UART 通信方式。本文档对 BMC81M001 的 Arduino Lib 函数、Arduino Lib 安装方式进行说明;范例演示了 TCP 和阿里云平台以及 ThingSpeak 的数据传输。

Arduino Lib 函数

Arduinc	Lib 名称: B	MC81M001		Lib 版本:	V1.0.3		
		构造函数	& 初始化				
	BMC81M001(HardwareSerial*theSerial1 = &Serial)						
	描述	构造函数,选择硬件 UAR	T 通信方式				
1	参数	* theSerial: 硬件 UART 通	信接口选择				
	返回值		—				
	备注		—				
	BMC81M001	(uint16_t rxPin,uint16_t txP	in)				
	描述	构造函数,选择软件 UAR	T 通信方式				
2	参数	rxPin: 软件 UART 的 RX txPin: 软件 UART 的 TX	引脚 引脚				
	返回值		—				
	备注		—				
	void begin(ui	$nt32_t baud = BMC81M001_t$	_baudRate)				
	描述	模块初始化					
3	参数	baud: 波特率, 默认 1152	00				
	返回值	void					
	备注		—				
		功能函数	牧 (TCP)				
	bool connect	ToAP(String ssid,String pass))				
	描述	模块连接 WiFi					
1	参数	ssid: WiFi 名称 pass: WiFi 密码					
-	返回值	执行情况: true: 成功 false: 失败					
	备注						
	bool connectTCP (String ip,int port)						
	描述	接入 TCP 服务器					
5 -	参数	ip: TCP 服务端的 IP 地址 port: 端口号					
	返回值	执行情况: true:成功 false:失败					
	备注	IP 地址和端口号由服务器	定义				



	bool writeDataTcp(int Dlength, char Dbuffer[])				
	描述	向 TCP 服务器发送数据			
	会粉	Dlength: 数据长度			
6	少奴	Dbuffer[]: 数据			
		执行情况:			
	返回值	true: 成功			
		false:			
	奋往 Ctaring and ID:				
	String readDa 世法	ata rep () 本接 TCD 肥久器 诗取肥久器台送的粉握			
7	加火 金粉	E按 ICP 加分奋, 以			
/		—————————————————————————————————————			
		ICP 服务奋反达的数据			
					
	haalaanfaM	功能函数 (10t)			
	server port).	qu(String chenuld,String username,String password,String mqu_nost,int			
	描述	配置 MOTT 连接参数			
	2,111	clientlid: 客户端、用户 ID。			
	参数	username: 用户名称。			
8		password: 密码。			
0		mqtt_host: 服务器地址。			
		server_port: 服务器端口。			
	返回值	执行情况:			
		false: 失败			
	备注				
	bool setPublis	shTopic(String publishtopic);			
	描述	设置发布的默认 Topic			
	参数	IoT 平台上 Topic 列表中的默认 publish Topic			
9		执行情况:			
	返回值	true: 成功			
		false: 失败			
	备注	—			
	cribetopic(String subscribetopic);				
	描述	设置订阅的默认 Topic			
	参数	IoT 平台上 Topic 列表中的默认 subscribe Topic			
10		执行情况:			
	返回值	true: 成功			
	友计				
		—			



	bool setTopic(String topic)						
	描述	设置 IoT 平台的自定义 topic					
	参数	topic: 消息发布者和订阅者之间的传输中介					
11	返回值	执行情况: true: 成功 false: 失败					
	备注	—					
	bool writeStr	ing (String Dbuffer,String topic)					
	描述	采用 String 数据格式向 IoT 平台发送数据					
12	参数	Dbuffer: 要发送的字符型数据 topic: 消息发布者和订阅者之间的传输中介					
12	返回值	执行情况: true: 成功 false: 失败					
	备注	—					
	bool writeBy	tes (char Dbuffer[],int Dlength,String topic)					
	描述	采用字节数据格式向 IoT 平台发送数据					
13	参数	Dbuffer[]: 数据 Dlength: 数据长度 topic: 消息发布者和订阅者之间的传输中介					
	返回值	执行情况: true: 成功 false: 失败					
	备注	—					
	void readIotI	Data (String *ReciveBuff,int *ReciveBufflen,String *topic)					
	描述	连接 IoT 平台后,读取平台发送的数据					
14	参数	*ReciveBuff: 接收到的数据 *ReciveBufflen: 接收数据长度 *topic: 接收到的数据所属 Topic					
	返回值	void					
	备注	—					
		其他函数					
	bool reset(vo	id)					
	描述	软件复位模块					
	参数	void					
15	返回值	执行情况: true: 成功 false: 失败					
	备注	—					



	int sendATC	ommand(String StringstrCmd,int timeout,uint8_t reTry);
	描述	发送 AT 指令
16	参数	StringstrCmd: AT 指令 timeout: 超时时间 reTry: 超时次数
	返回值	执行情况: true: 成功 false: 失败
	备注	—

注 1: 登录阿里云物联网平台,进入控制台→选择公共实例→设备管理→设备→点击查看进入对应的 产品→点击 MQTT 连接参数查看即可。

设备管理 ^	设备									
产品	#1600开关 ~	©≌#### € 1	9	• 20309 0 1		■ 用約在現				
0# 1	设备列表 此次管理	高级搜索								
94B	151020 Hallioto	DeviceName V	遺稿入 DeviceName	Q	请达/平众加	5% V				
设备模拟器	DeviceName/酸注名称		设备所属产品	节点类型		22005 O 7	5	統上總时间	庶用/禁用	
设备分发	Arduino_Wifi		暂能的开关	(2 9)		• EK	21	023/02/16 15:59:01.886		
IoT李生引擎 New	物联网亚本		Arduino Mifi							
实例详情	1040131									
设备管理		rduino_W	ifi œ							
产品	产品	智能的开关	查看						DeviceSecret	****** 查看
		hj3hM0mUh	iMj 复制							
设备	设备信	Topic 列表	物模型数据	设备影子	文件管:	理 日志服务	在线调试	分组		
分组	10.00									
设备模拟器	设备信息									_
	产品名称	智能	的开关			MQTT 连接参数				×
设备分发	节点类型	· (2)新				clientId	hj3hM0mUhMj.Ar	duino_Wifi securemode=2,	signmethod=hmacsha256,times	tamp=1676
IoT孪生引擎 New	開注合が「	2022	0/11/22 12:42:57				535334057			
	いたりに	2022	:/11/22 15:45:57			username	Arduino_Wifi&hj3	hM0mUhMj		
消息转及		9 U.A. 5%7			1	passwd	0665e145e05a07e	c63274d66766db4a64aaa4	7d537f8af4c1350f5de67c37670	
监控运维	< MQI DER	2 50 宣君	4		1	mqttHostUrl	iot-06z00ac1cwfk	n1s.mqtt.iothub.aliyuncs.co	m	
安全中心					5	port	1883			
	设备扩展	息							一键复制	关闭
仿真实验	SDK 语言									



注 2: 登录阿里云物联网平台,进入控制台→选择公共实例→设备管理→产品→点击查看进入对应的 产品→Topic 类列表→自定义 Topic

いたのでは	~	← 智能的	开关 ^{发布}	
以田邑庄		ProductKey ł	hj3hM0mUhMj 复制	
产品		设备数	1 前往管理	
设备		产品信息 Top	pic 类列表 功能	能定义
分组			ゆ 방 귀 나 중 / 는 ㅜ	
设备模拟器		基础通信 lopic	物煤空通信 10p	
设备分发		定义 Topic 类		
loT孪生引擎 New		自定义 Topic		
消息转发	~	/hj3hM0mUhMj/\${de	eviceName}/user/ardu	nio
监控运维	~ <	/hj3hM0mUhMj/\${de	eviceName}/user/upda	ate publishtopic
安全中心	\sim	/hj3hM0mUhMj/\${de	eviceName}/user/upda	ate/error
仿真实验				
文档与工具		/hj3hM0mUhMj/\${de	eviceName}/user/get	subscribetopic



Arduino Lib 下载及安裝

BMC81M001 Library: 可参考下面两种方法安装 BMC81M001 的 Arduino Library

方式1: 搜索安装

捜索安装: Arduino IDE → 项目 → 加载库 → 管理库 → 捜索 BMC81M001 → 安装

	and an advantage of the state	-1815		
文件编辑项目	目 工具 帮助			
	验证/编译	Ctrl+R		
	上传	Ctrl+U		
an arright	使用编程器上传	Ctrl+Shift+U	a manual a	
8/***	导出已编译的二进制文件	Ctrl+Alt+S		
711+	显示项目文件夹	Ctrl+K	DistanceDet.	a. 1.00
	加载库	3	Δ	
	添加文件		管理库	Ctrl+Shift+I
2011.0			添加 .ZIP 库	方式 1

搜索安装流程1

◎◎ 库管理器	×
类型 全部 主題 全部 BMC81M001	
DBT122 sease blowy by Renew Nicohimeti BHT22 hours a Nicohimeti BHT22 hours a Nicohimeti annual of the Nicohimetic Sease and State of the Nicohimetic Sease and State of Andore Sease and State of Andore Sease and State of Andore Sease and Andore Se	^

搜索安装流程2

方式 2: 添加.ZIP 库, 需提前下载.ZIP 库

下载方法:打开倍创官方网站 (<u>https://www.bestmodulescorp.com/bmc81m001</u>. <u>html</u>) 文件目录下的 Arduino 范例程序 (BMC81M001 Library)。 添加.ZIP 库: Arduino IDE → 项目 → 加载库 → 添加.ZIP 库 ...

					
文件 编辑	项目	工具 帮助			
Ø 0		验证/编译 上传 使用编程器上传	Ctrl+R Ctrl+U Ctrl+Shift+U		
6/**		导出已编译的二进制文件 显示项目文件来	Ctrl+Alt+S		
		加载库)	Δ	
24.0		添加文件		管理库	Ctrl+Shift+I
			. Software	添加.ZIP 库	
			hards	Arduino 库	方式2



Arduino 范例

范例 1: TCP



实物连接示意图

范例 1 实现功能:模块连接手机热点,实现与手机上的 TCP 服务端 APP 双向数据通信。

1. 范例打开方式:

Arduino IDE → 文件 → 示例 → Lib 选择 (BMC81M001) → 选择范例 (TCP)

2. 示例说明:

首先需要在 TCP.h 文件中修改 WiFi 信息和 TCP 服务器信息如下:







3. 连接 WiFi, 接入 TCP 服务器

```
#include "TCP.h"
BMC81M001 Wifi (6,7);
void setup()
{digitalWrite(LED, LOW);
 Serial.begin(9600);
                           // 配置串口监视器
 Wifi.begin();
                            // 初始化模块
 if(!WiFi.connectToAP(WIFI_SSID,WIFI_PASS)) // 设置连接的热点名称和密码
 {
   Serial.println("Disconnect to WIFI");
  }
 if(!WiFi.connectTCP(IP,IP_Port)) // 连接到 TCP 服务器 (APP)
  {
   Serial.println("Disconnect to TCP server");
  }
 else
  {Serial.println("Connected to TCP"); }
```

4. 在此范例中,在串口监视器发送数据,将直接上传到 TCP 服务器,在 TCP/ UDP 网络调试助手上可以看到模块发送的信息;在 TCP/UDP 网络调试助手 上发送的数据可以在串口监视器查看。

```
void loop() {
tcpBuff=Wifi.readDataTcp(); // 监听模块接收的数据
if(tcpBuff!=0)
{
 Serial.println(tcpBuff);
}
while (Serial.available() > 0) // 接收到监视串口发送的数据,透传发送数据
{
 SerialBuff[resLen++] = mSerial.read();
 delay(10);
}
if(resLen>0)
{
 digitalWrite(LED, HIGH); // 有数据亮灯
 if(WiFi. writeDataTcp (resLen, SerialBuff)) // 向 TCP 服务器发送数据
  {
   Serial.println("Send data sucess");
   digitalWrite(LED,LOW); // 数据发送成功灭灯
  }
 clearBuff();
}
void clearBuff() {
 memset(SerialBuff,'\0', RES MAX LENGTH);  // 清空接收串口数据的字符型数组
  resLen = 0;
```



程序执行结果在 APP 和串口监视器结果显示:

= Port:6000	∞ COM4	– 🗆 X
		发送
1234	10:00:20.972 -> Connected to TCP	
_	10:00:30.152 -> 1234	
4567	10:00:30.152 ->	
4007	10:00:46.614 -> Send data sucess	
		10.0389 2000244444

范例 2: Alinyun_Iot



实物连接示意图

范例2实现功能:模块连接手机热点,连接阿里云平台上的实例,实现数据透传功能。

1. 范例打开方式:

Arduino IDE → 文件 → 示例 → Lib 选择 (BMC81M001) → 选择范例 (Alinyun_ Iot)

2. 登录物联网平台 (aliyun.com) 注册后选择公共实例,创建产品及设备,自定 义Topic,将MQTT参数以及各类Topic填入Aliyun_Iot.h文件,具体流程如下:



	物联网平台 / 设备管理 / 产品 / 新建产品帮助文档
实例详情	
设备管理	
产品 1	新建产品从设备中心新建产品
设备	* 产品名称
分组	smart (2)
设备模拟器	 * 所屬品类 ● ○ 标准品类 ● 自定义品类 3
设备分发	* 节点美型
IoT孪生引擎 New	
消息转发	
监控运维	☆ / 注网与数据
安全中心	* 连网方式
仿真实验	Wi-Fi 🗸
立档与工具	* 数据格式 @
X11512	ICA 标准数据格式 (Alink JSON) V
	◆ 校验类型
	◇认证方式
	更多信息
	◇产品描述
	<u>(</u>)
曰 新版反馈	确认 取消

1) 创建产品: 进入公共实例后, 左侧菜单设备管理 \rightarrow 产品 \rightarrow 创建产品

2) 自定义 Topic,定义数据传输中介:设备管理 → 产品 → 查看刚刚新建的 产品 → Topic 类列表 → 自定义 Topic

实例详情	~	← 智能的开关 🧯		
≠品 1		ProductKey hj3hM0mUhMj 复制 设备数 1 前往营理	定义 Topic 类	×
设备 分组 设备模拟器		产品信息 Topic 23 功能定义 消息解析 服务编订阅 基础通信 Topic 物模型通信 Topic 自定义 Topic 3	Topic 格式必须以广进行分量、区分每个进目、其中前三个类目已起现 进好、算一个代数带插标但 Productky, 算一个 SideviceNamei 通配 DeviceName, 第三个 vace 用标识可不图的意义 Topic 就, 简单来 说, Topic 挫, / a151****dhX/SideviceName/Juser/update 是具体 Topic / / a151****dhX/mydevice/Juser/update 是具体 / a151****dhX/mydevice/Juser/update 御訳会,	
设备分发 IoT孪生引擎 New		定义 Topic 4 超定义 Topic	* 设备操作反限 发布和订阅 (5)	~
消息转发	~	/hj3hM0mUhMj/\${deviceName}/user/ardunio	* Торіс 英	
监控运维	~ <	/hj3hM0mUhMj/\$(deviceName)/user/update	/hj3hM0mUhMj/\${deviceName}/user/arduino arduino	
安全中心		/hj3hM0mUhMj/\$[deviceName]/user/update/error	描述	
「「東東拉		/hj3hM0mUhMj/\${deviceName}/user/get	请编入描述	
又档与上具			0/1 7 #64	100 単





回到"产品"界面→"管理设备"→"添加设备"→ DeviceName, 点击确认。

	产品	(设备模型)						
		(,	Market Later & Salar West Later Life					
			设备接入流程概览					
分组		联州平田 军民使用派示	01 创建产品 产品是同品类设备的集合	02 创建设备 创建设备获取注册平台所需的	03 编辑物理型 产品下的设备都会继承产品的	04 设备端开发 集成Link SDK, 开发设备续程	05 查看上报数据 查看设备上报的圈性数据 SDK,开发	
设备模拟器				9601B/8	404522	75	(CHESSING) P	
设备分发	632768	快速入门 清编入7	*品名称查询 Q	清选项产品标签 🗸				
IoT字生引擎 New	产品名称		ProductKey	节点类型	创建的订订		题作 2	
消息转发 ~	都能的开关	ŧ	hj3hM0mUhMj	(2 %	2022/11/22 13:39:28		查看 管理设备 删除	
	^	设备						
设合官理				设备总数 🕗	● 激活设备 @	 当前在线 @ 		
产品		智能的并关	×	1	1	1		
设备		设备列表 批	次管理 高级搜索				_	
分组		NEADVIDER 2		1440 0 1 11	添加设备 ❷		×	
设备模拟器			連790月 Devicemame	✓ Iŋ細八 DeviceNami	e	DeviceName 可以为空,当为空时	1. 阿里云会颁发产品下的	
边条公告		DeviceName/1	备注名称	设备所属产品	唯一标识符	作为 DeviceName。		
设置刀汉		Arduino_Wifi		智能的开关	产品			
IoT孪生引擎 New					智能的开关			
消息转发	~				DeviceName 👩	_		
监控运维	~ <				Arduino-Wifi	(4)		
安全中心					备注名称 🕝	Ŭ		
44-17-14-34					请输入借注名称			
1万具头拉							(5)	
文档与工具							确认 取消	

4) 查看设备的 MQTT 连接参数:设备 → 查看刚刚创建的设备 Arduino-Wifi → 设备信息 → MQTT 参数查看;查看设备的 Topic 同步骤 2 中的自定义 Topic 步骤相同。

将 CLIENTLID、USERNAME、PASSWORD、MQTT_HOST、SERVER_ PORT 等 MQTT 参 数 填 入 Aliyun_Iot.h 文 件 里,将 PUBLISHTOPIC、 SUBSCRIBERTOPIC、CUSTOMTOPIC 等类 Topic 也填入 Aliyun_Iot.h 文 件里。





Allowed Lines - Allowed Lines - Allowed Lines						
1 #ifndef BMC81M001 H	MQTT 连接参数					×
2 #define _BMCS1M001_H	clientId	hj3hM0mUhMj. 535334057	Arduino_Wifi securer	node=2,signmethod=h	macsha256,time	stamp=1676
	username	Arduino_Wifi&h	nj3hM0mUhMj			
5 //*********************** Header *****************//	passwd	0665e145e05a0)7cc63274d66766db4	a64aaa47d537f8af4c13	350f5de67c37670)
6 //***********************************	mattHostUrl	iot-06z00ac1cw	/fkn1s.matt.iothub.ali	/uncs.com		
7 #include "Aliyun_Iot.h"	port	1883				
9						
10 //***********************************					一键复制	关闭
11 //*********** wifi information *************//	~~~~		414	***	13-4-	
12 //***********************************	设备管理	~	← 笛崩	的开大	反仰	
14 #define WIFI SSID "iQOO Neo 855			ProductKey	hj3hM0mUhM	复制	
15 #define WIFI_PASS "12345678 !"	产品		设备数	1 前往管理		
	设备					
1/ #define CLIENTLID "mytest[securemode=3\signmethod=nr 18 #define USERNAME "Arduino Wifisggzn81RWZC2"	60C (MI		产品信息	Topic 关列表	功能定义	
19 #define PASSWORD "9CF1D3 420F07ECC02250EF829D9EAC8529A(分组					_
20 #define MQTT_HOST "gqzn81RWZC2.iot-as-mqtt.cn-shanghai.	「たち」またい「月月		基础通信下	opic 物模型通	信 Topic	É
21 #define SERVER_PORT 1883	设畲侯扒醋					
23 #define PUBLISHTOPIC "gqzn81RWZC2/Arduino_Wifi/user/upc	设备分发		포文 10016 34			
24 #define SUBSCRIBERTOPIC "ggzn81RWZC2/Arduino_Wifi/user/ 25 #define CUSTOMTOPIC "ggzn81RWZC2/Arduino_Wifi/user/ardu	loT变生引整 N	ew	自定义 Topic			
24 	1017 1314		(h:2hM0m10	Miller in Name) (istomtonic
	消息转发	\checkmark	/njsniviomor	ivij/s(deviceivarite)/d		
	监控运维	~	< /hj3hM0mUł	Mj/\${deviceName}/u	ser/update <mark>PU</mark>	Iblishtopic
	安全中心	~	/bi2bM0mLlk	Mi/\$/deviceName)/w	rar/updata/arro	-
	仿真实验		, njoninon or	ing stachechancy a	sel) apaate) en o	
			/hj3hM0mUh	Mj/\${deviceName}/u	ser/get sub	scribetopic
	文档与工具					
// 连接的 wi ri 的账户家码						
	115	ᆂ	5 46			
#define WIFI_SSID "1Q00_Neo_855"	// 于	化恐思剂	百个小			
#define WIFI PASS "12345678.!"	// 手	机热点图	密码			
// alivun MOTT 信息						
ha fine OI TENELTE Handersteller			- la a - la la su		`	
#denne CLIENTLID "mytest securemo	ae=3\s	lgnme	tnoa=nm	acsnal \	\ <i>,</i>	
timestamp=6789	"					
#define USERNAME "Arduino Wifi&gazn	81RWZC2"					
#define PASSWORD "9CF1D3420F07ECC0	2250EF82	9D9EA	C8529A8	D672"		
#define MOTT HOST "gaznelDM7C2 jot	-20-ma++		hanghai	 ລີ່] ເຫຼາກ:		, ''
#define Myri_nosi yqzhorKW2C2.100	аз-шүсс ,, <u>з</u> ш		nanynat	.arryund	La. COI	.L
#deline SERVER_PORT 1883	//	口丂				
// aliyun TOPIC 信息						
#define PUBLISHTOPIC "gqzn81RWZC2/	Arduino	Wifi/us	ser/upd	ate"		
#define_SUBSCRIBERTOPIC_"ggzn81RWZC2/Arduino_Wifi/user/get"						
#define CUSTOMTOPIC "ggzn81PWZC2/A	rduino M	1i fi / 11 6/	or/ardu	nio"		
#denne costomioric gdznotkwzcz/Arduino_win/user/ardunio~						

注意: CLIENTID 填写时需留意在","前需加\\。



3. 示例说明:

根据阿里云平台产生的数据,对模块进行初始化。其中包括软件复位模块, 接入WiFi,接入阿里云 IoT,如下:

```
#include "Aliyun Iot.h"
BMC81M001
              Wifi(&Serial1);
void setup()
{
  digitalWrite(LED, LOW);
                                // 配置串口监视器
  Serial.begin(9600);
  Wifi.begin();
                                 // 模块初始化及配置
  Wifi.reset();
  Serial.print("WIFI Connection Results: ");
  if(Wifi.connectToAP(WIFI_SSID,WIFI_PASS)==0) // 根据名称和密码连接WiFi
  {
    Serial.println("fail");
  }
  else {Serial.println("success");}
  Serial.print("Aliyun Connection Results: ");
  Wifi.sendATCommand("AT+CIPSNTPCFG=1,8,\"ntp1.aliyun.com\"",1000,2);
  // 连接阿里云
  if (Wifi.configMqtt (CLIENTLID, USERNAME, PASSWORD, MQTT HOST,
     SERVER PORT)==0) // 配置 MQTT 参数
  {
    Serial.println("fail");
  }
  else
  {
    Serial.println("success");
   Wifi.setPublishTopic(PUBLISHTOPIC); // 连接成功后配置订阅和发布 Topic
   Wifi.setSubscribetopic(SUBSCRIBERTOPIC);
  }
  Serial.print("Topic set Results: ");
  if (Wifi.setTopic (CUSTOMTOPIC) == 0) // 设置自定义 Topic
  {
    Serial.println("fail");
  }
  else {Serial.println("success");}
  delay(200);}
```

4. 运行后,即可在平台上查看到设备状态。

DeviceName/备注名称	设备所属产品	节点类型	设备状态 🚱 🖓	最后上线时间	启用/禁用	操作
Arduino_Wifi	智能的开关	设备	 在线 	2023/02/16 15:59:01.886		查看 删除



在此范例中,当串口监视器发送数据,模块只做透传功能,将接收到串口监视器发送的数据直接上传到云平台,可以在平台日志记录查看到模块上传的信息。

```
void loop()
{
 Wifi.readIotData(&aliyunReciveBuff, &aliyunReciveBufflen, &recTopic);
// 监听模块接收的数据
 if (aliyunReciveBufflen)
  {
   Serial.println(aliyunReciveBufflen);
   Serial.println(aliyunReciveBuff);
  }
// 接收到监视串口发送的数据,透传发送数据
                                  // 接收监视串口发送的数据
 while (Serial.available() > 0)
  {
   SerialBuff[resLen++] = Serial.read();
   delay(10);
  }
  if(resLen>0)
  {
   digitalWrite(LED, HIGH);
   DATA BUF = (String )SerialBuff;
   topic = PUBLISHTOPIC;
   if(Wifi.writeString(DATA BUF, topic))
    {
      Serial.println("Send String data sucess");
    if(Wifi.writeBytes(SerialBuff,resLen,topic))
    {
     Serial.println("Send byte data sucess");
    }
   clearBuff();
  }
}
void clearBuff() {
 memset(SerialBuff,'\0',RES MAX LENGTH); // 清空数据接收数组
 resLen = 0;
```



范例 3: ThingSpeak



实物连接示意图

范例 3 实现功能:模块连接手机热点,连接 ThingSpeak 平台上的通道,发送数据到平台上显示。

- 1. 范例打开方式: Arduino IDE → 文件 → 示例 → Lib 选择 (BMC81M001) → 选择范例 (ThingSpeak)
- 2. 登录 ThingSpeak 平台 (<u>https://thingspeak.com/</u>), 注册账号后, 新建数据通道, 再新建 MQTT 设备, 将 MQTT 参数填入 ThingSpeak.h 文件中。
 - 1) 注册账号并进入平台:点击 Get Start For Free,进入平台界面,注册账号 并登入。



2) 创建 channels: 点击 New Channel, 填入 Name 和开启数据字段数量, 保存 通道即可。

C ThingSp	eak™	New Char	nel		
Signed in successfully.		×	Name		
My Cha	nnels		Description		
New Channe Search by tag		٩	Field 1 Field Label 1		
Name 🗢	Updated 🗢		Field 2		
A Wifi	2023-06-08 05:57		Field 3		
			1		
			Save Channel		

3) 新建 MQTT 设备:填入设备名称 Wifi,选择新建的通道 Wifi,点击 Add channel,显示 Allow Publish 和 Sublish OK 后 Add Device。

□ , ThingSpeak [™]	□ ThingSpeak [™]	
Channels +	MQTT Devices	
Apps •	dd a new device	Authorize channels to access 🛛
Devices •		Select a Channel V
MQTT	+	
Support •	Add a new device ×	
How to Buy	Device Information	Add Channel
	Name*	Authorized Allow Allow Channel Dublish Subscribe
	Wifi	Wifi (2181120)
	Description	
	Enter optional information about this device for later reference.	Cancel Add Device
	Authorize channels to access ()	
	Wifi (2181120)	
	Add Channel	
	Authorized Allow Allow	
	Channel Publish Subscribe	
	No channels authorized.	
	Cancel Add Davice	
	Carlor Add Device	



4) 点击 edit, 查看新建 MQTT 设备的 MQTT 参数,将数据填入 ThingSpeak.h 文件中。





3. 示例说明:

模块初始化后,连接热点,配置 MQTT 参数,连接 ThingSpeak 平台的通道,发送数据至平台。如下:

```
#include "ThingSpeak.h"
BMC81M001
             Wifi(&Serial1);
void setup()
{
  digitalWrite(LED, LOW);
  Serial.begin(9600); // 配置串口监视器
  Wifi.begin();
                           // 模块初始化及配置
 Wifi.reset();
  Serial.print("WIFI Connection Results: ");
  if(Wifi.connectToAP(WIFI_SSID,WIFI_PASS)==0) // 根据名称和密码连接WiFi
  {
    Serial.println("fail");
  }
  else {Serial.println("success");}
  Serial.print("ThingSpeak Connection Results: ");
  if (Wifi.configMqtt(CLIENTLID, USERNAME, PASSWORD, MQTT HOST,
     SERVER PORT)==0) // 配置 MQTT 参数
  {
    Serial.println("fail");
  }
  else
  {
    Serial.println("success");
  }
  delay(200);}
```



 在此范例中,在串口监视器发送数据,数据会直接上传到云平台,可以在通 道接口看到数据。

```
void loop()
{
 Wifi.readIotData(&ReciveBuff, &ReciveBuffen, &recTopic); // 监听模块
                                                 // 接收的数据
 if (ReciveBufflen)
 {
   Serial.println(ReciveBufflen);
  Serial.println(ReciveBuff);
 }// 接收到监视串口发送的数据,透传发送数据
                                      // 接收监视串口发送的数据
 while (Serial.available() > 0)
  {
 SerialBuff[resLen++] = Serial.read();
 delay(10);
  }
 if(resLen>0)
 {
   digitalWrite(LED, HIGH);
   DATA BUF = "field1=";
                                     // 发送通道字段1数据
   DATA_BUF += SerialBuff;
   topic = PUBLISHTOPIC;
   if(Wifi.writeString(DATA_BUF,topic))
   {
     Serial.println("Send String data sucess");
   }
   clearBuff();
   digitalWrite(LED, LOW);
   }
 }
void clearBuff() {
 resLen = 0;
```



5. ThingSpeak 显示结果,可以从图表中看到数据的接收。

Private View	Public View	Channel Settings
Sharing A	PI Keys Data	a Import / Export
🗄 Add Visua	alizations	+ Add Widgets
Export red	cent data	
hanne	Stats	
		7





范例 4: ThingSpeakPublish



实物连接示意图

范例 4 和范例 5 可搭配使用,实现两个 WiFi 设备的远程数据交换,一个作为 Publish 端,另一个作为 Subscribe 端。实现功能如下:

Publish 端:将温湿度模块 BME33M251 的温度、湿度数据上传到 ThingSpeak; Subscribe 端:从 ThingSpeak 读取 Publish 端刚上传的温度、湿度数据,并显示于 OLED 模块上;





范例 4 实现功能:读取温湿度模块 BME33M251 的温度、湿度数据,通过 MQTT 设备 1 (Publish)发布到 ThingSpeak 平台的通道中。

- 1. 范例打开方式: Arduino IDE → 文件 → 范例 → Lib 选择 (BMC81M001) → 选 择范例 (ThingSpeakPublish)。
- 2. 登录 ThingSpeak 平台 (<u>https://thingspeak.com</u>), 注册账号后, 新建数据通道 (wifi t&h), 新建 MQTT 设备 (Publish), 将 MQTT 参数填入 ThingSpeakPublish. h 文件中。



1) 注册账号并进入平台:点击 Get Start For Free,进入平台界面,注册账号并登入。



2) 创建 channels: 点击 New Channel, 填入自定义 Name (例 wifi t&h) 和 Field1 (例 humidity)和 Field2 (例 temperature), 保存通道即可。

C ThingSp	eak™	New Cha	annel		
Signed in successfully.		X	Name		
My Cha	nnels		Description		
New Channe Search by tag	4	Q	Field 1 Field Label 1		
Name 🗢	Updated \$		Field 2		
Wifi	2023-06-08 05:57		Field 3		
					
			Save Channe	əl	



3) 新建 MQTT 设备 1: 填入设备名称 (Publish),授权通道 wifi t&h 的发 布和订阅,点击 Add channel,显示 Allow Publish 和 Sublish OK 后 Add Device。此 Publish 设备用作上传温湿度模块数据。

ิ, ThingSpeak [™]	□ □ ThingSpeak ™	
Channels • Apps •		Authorize channels to access ①
MQTT Support - Commercial Use	Edit Subscribe	
How to Buy	Device Information	Authorized Allow Allow Channel Publish Subscribe
	Name Subscribe	Wiff (2181120)
	Description Enter optional information about this device for later reference.	Cancel Add Device
	MQTT Credentials Use these MQTT credentials to publish and	Add a new devic
	Subscribe to ThingSpeak channels. Learn Mor	Device Details: Publish Edit
	LiwJDBccDxULGxUwDiANEjs	No description Delete
	Username	Authorized Channels and Permissions:
	LiwJDBccDxULGxUwDiANEjs	Wifi t&h (2181120)
	Password	
	IG	

4) 点击设备 (Publish) 右侧的 "edit" 按钮, 查看 MQTT 设备 (Publish) 的 MQTT 参数, 将数据填入 ThingSpeakPublish.h 文件中。





3. 示例说明:

模块初始化后,连接热点,配置 MQTT 参数,连接 ThingSpeak 平台的通道,如下:

```
#include "ThingSpeak.h"
#include <BMD31M090.h>
                                              // 温湿度模块
#include <BM25S2021-1.h>
BMC81M001 Wifi(&Serial1);
BM25S2021 1 BMht(&Wire2);
void setup()
{
  Serial.begin(9600);
  BMht.begin();
                                              //WIFI 模块、温湿度模块初始化
  Wifi.begin();
  Wifi.reset();
  Serial.print("WIFI Connection Results:");
  if(Wifi.connectToAP(WIFI SSID,WIFI PASS)==0) // 连接热点
  {
    Serial.println("fail");
  }
  else {Serial.println("success");}
  Serial.print("ThingSpeak Connection Results:");
if (Wifi.configMqtt(CLIENTLID, USERNAME, PASSWORD, MQTT HOST, SERVER PORT) == 0)
// 根据 MQTT 连接 ThingSpeak 平台
  {
    Serial.println("fail");
  }
  else {Serial.println("success");}
  delay(200);
  Wifi.setPublishTopic(PUBLISHTOPIC); // 订阅 Topic · 即通道数据更新
  Wifi.setSubscribetopic(SUBSCRIBERTOPIC2);
  Wifi.setSubscribetopic(SUBSCRIBERTOPIC1);
  topic = PUBLISHTOPIC;
                                   // 发布 Topic
}
```



4. 轮询温湿度模块的数据,将湿度发送至通道中的 field1,将温度发送至通道的 field2。

```
void loop()
{
 data1=BMht.readHumidity(); // 获取湿度数据
Humidity=String(data1,2); // 译点型数据转成字符串
// 诵道 1 数据
 DATA BUF = "field1=";
                                   // 通道1数据
 DATA BUF += Humidity;
 if (Wifi.writeString (DATA BUF, topic)) // 发布数据至通道1
   {
     Serial.println("Send String data sucess");
     delay(1000);
   }
 clearBuff();
                                     // 获取温度数据
 data1=BMht.readTemperature(false);
 Temperature=String(data1,2);
                                          // 浮点型数据转成字符串
 DATA BUF = "field2=";
                                          // 通道 2 数据
 DATA BUF += Temperature;
 if(Wifi.writeString(DATA_BUF,topic)) // 发布数据至通道 2
  {
     Serial.println( "Send String data sucess" );
     delay(1000);
   }
 clearBuff();
 delay(2000);
 }
```





范例 5: ThingSpeakSubscribe



实物连接示意图

范例 5 实现功能:订阅数据更新 Topic,通过新建并连接 MQTT 设备 2 (Subscribe)接收 ThingSpeak 对应通道最新发布的温湿度数据,并显示在 OLED 模块上。

- 1. 范例打开方式: Arduino IDE → 文件 → 范例 → Lib 选择 (BMC81M001) → 选择范例 (ThingSpeakSubscribe)。
- 2.ThingSpeak 通道无需新建,使用范例4的wifit&h。下面新建MQTT设备 (Subscribe)以及填写参数至ThingSpeakSubscribe.h文件。
 - 1) 新建 MQTT 设备 2: 填入设备名称 (Subscribe), 授权通道 wifi t&h 的发 布和订阅,点击 Add channel,显示 Allow Publish 和 Sublish OK 后 Add Device。此 Subscribe 设备用于接收通道最新更新的数据。





4) 点击设备 (Subscribe) 右侧的 "edit" 按钮, 查看 MQTT 设备 (Subscribe) 的 MQTT 参数, 将数据填入 ThingSpeakSubscribe.h 文件中。





3. 示例说明:

模块初始化后,连接热点,配置 MQTT 参数,连接 ThingSpeak 平台的通道。如下:

```
#include "ThingSpeakRead.h"
#include <BMD31M090.h>
#include <BM25S2021-1.h>
                                            //OLED 模块
BMC81M001 Wifi(&Serial1);
BMD31M090
             BMD31(128,64, &Wire2);
float data1;
String Humidity;
String Temperature;
void Display();
void setup()
{
  BMD31.begin(0x3C);
  BMD31.setFont(FontTable_8X16);
  delay(100);
  Serial.begin(9600);
  Wifi.begin();
  Wifi.reset();
  Display();
  Serial.print("WIFI Connection Results:");
  if(Wifi.connectToAP(WIFI SSID,WIFI PASS)==0) // 连接热点
  {
    Serial.println("fail");
  }
  else {Serial.println("success");}
  Serial.print("ThingSpeak Connection Results:");
if (Wifi.configMqtt(CLIENTLID, USERNAME, PASSWORD, MQTT HOST, SERVER PORT) == 0)
// 根据 MQTT 连接 ThingSpeak 平台
  {
    Serial.println("fail");
  }
  else {Serial.println("success");}
  delay(200);
                                            // 订阅 Topic · 即通道数据更新
  Wifi.setPublishTopic(PUBLISHTOPIC);
  Wifi.setSubscribetopic(SUBSCRIBERTOPIC2);
  Wifi.setSubscribetopic(SUBSCRIBERTOPIC1);
```



4. 轮询平台是否有更新数据,有数据则识别对应数据,显示在 OLED 模块上。

```
void loop()
{
 Wifi.readIotData(&ReciveBuff,&ReciveBufflen,&topic); // 接收平台下发的数据
 if(ReciveBufflen)
  {
                                                   // 通道1数据为湿度
   if(topic==SUBSCRIBERTOPIC1)
    {
     ReciveBuff.toCharArray(OledBuff,7);
     BMD31.drawString(40,2,(u8*)OledBuff);
                                                 //OLED 显示
     BMD31.drawString(85,2,(u8*)"%");
     ReciveBufflen=0;
    }
                                                   // 通道 2 数据为温度
   if(topic==SUBSCRIBERTOPIC2)
    {
     ReciveBuff.toCharArray(OledBuff,7);
     BMD31.drawString(40,6,(u8*)OledBuff);
                                                //OLED 显示
     BMD31.drawString(85,6,(u8*)"C");
     ReciveBufflen=0;
    }
  }
                                                   // 清空数据
  clearBuff();
```

5.OLED 显示结果如下:





Copyright[©] 2023 by BEST MODULES CORP. All Rights Reserved.

本文件出版时倍创已针对所载信息为合理注意,但不保证信息准确无误。文中提到的信息仅是提供 作为参考,且可能被更新取代。倍创不担保任何明示、默示或法定的,包括但不限于适合商品化、 令人满意的质量、规格、特性、功能与特定用途、不侵害第三方权利等保证责任。倍创就文中提到 的信息及该信息之应用,不承担任何法律责任。此外,倍创并不推荐将倍创的产品使用在会由于故 障或其他原因而可能会对人身安全造成危害的地方。倍创特此声明,不授权将产品使用于救生、维 生或安全关键零部件。在救生/维生或安全应用中使用倍创产品的风险完全由买方承担,如因该等 使用导致倍创遭受损害、索赔、诉讼或产生费用,买方同意出面进行辩护、赔偿并使倍创免受损害。 倍创(及其授权方,如适用)拥有本文件所提供信息(包括但不限于内容、数据、示例、材料、图形、 商标)的知识产权,且该信息受著作权法和其他知识产权法的保护。倍创在此并未明示或暗示授予 任何知识产权。倍创拥有不事先通知而修改本文件所载信息的权利。如欲取得最新的信息,请与我 们联系。