



XR806 MQTT 网络模块 开发指南

版本号：1.0

发布时间：2020-11-12

版本历史

版本	日期	责任人	版本描述
1.0	2020-11-12	AWA 1096	创建文档。

目录

版本历史.....	i
目录.....	ii
表格目录.....	2
1 前言.....	3
1.1 文档简介.....	3
1.2 目标读者.....	3
1.3 适用范围.....	3
1.4 文档约定.....	3
1.4.1 标志说明.....	3
2 概述.....	4
2.1 背景说明.....	4
2.2 规格特性.....	4
2.3 文件位置.....	4
3 应用说明.....	5
3.1 应用简述.....	5
3.2 配置说明.....	5
3.3 接口说明.....	5
3.3.1 初始化接口.....	6
3.3.1.1 NewNetwork.....	6
3.3.1.2 MQTTClient.....	6
3.3.2 连接接口.....	7
3.3.2.1 ConnectNetwork.....	7
3.3.2.2 TLSConnectNetwork.....	7
3.3.2.3 MQTTConnect.....	8
3.3.3 MQTTPublish 接口.....	10
3.3.3.1 MQTTPublish.....	10
3.3.4 MQTTYield 接口.....	10
3.3.4.1 MQTTYield.....	10
3.3.5 MQTTSubscribe 接口.....	11
3.3.5.1 MQTTSubscribe.....	11
3.3.6 MQTTUnsubscribe 接口.....	11
3.3.6.1 MQTTUnsubscribe.....	11
3.3.7 MQTTUnsubscribe 接口.....	12

3.3.7.1 MQTTDisconnect..... 12

4 示例说明..... 13

4.1 示例简介..... 13

4.1.1 获取方式..... 13

4.1.2 准备工作..... 13

4.1.3 操作步骤..... 13

4.2 效果展示..... 14

4.3 实现流程..... 14



表格目录

表 2-1	MQTT 模块的功能特性.....	4
表 2-2	MQTT 网络协议的文件位置.....	4
表 3-1	XR806 MQTT 模块配置列表.....	5
表 3-2	MQTT 模块常用接口简介.....	5
表 3-3	NewNetwork 接口函数说明.....	6
表 3-4	MQTTClient 接口函数说明.....	6
表 3-5	ConnectNetwork 接口函数说明.....	7
表 3-6	TLSCoconnectNetwork 接口函数说明.....	7
表 3-7	MQTTConnect 接口函数说明.....	8
表 3-8	MQTTPacket_connectData 结构体的成员说明.....	9
表 3-9	MQTTPublish 接口函数说明.....	10
表 3-10	MQTTYield 接口函数说明.....	10
表 3-11	MQTTSubscribe 接口函数说明.....	11
表 3-12	MQTTUnsubscribe 接口函数说明.....	11
表 3-13	MQTTDisconnect 接口函数说明.....	12
表 4-1	MQTT 示例工程的命令列表.....	13

1 前言

1.1 文档简介

本文档介绍了 XR806 平台上 MQTT 模块的使用方法。

1.2 目标读者

XR806 开发及维护人员

1.3 适用范围

此文档适用于 XR806 SDK，支持 XR806 系列芯片产品

1.4 文档约定

1.4.1 标志说明

本文档采用各种醒目的标志来表示在操作过程中应该特别注意的地方，这些标志的含义如下：

标识	说明
 警告	该标志后的说明应给予格外关注，如果不遵守，可能会导致人员受伤或死亡。
 注意	提醒操作中应注意的事项。不当的操作可能会损坏器件，影响可靠性、降低性能等。
 说明	为准确理解文中指令、正确实施操作而提供的补充或强调信息。
 窍门	一些容易忽视的小功能、技巧。了解这些功能或技巧能帮助解决特定问题或者节省操作时间。

2 概述

2.1 背景说明

MQTT (Message Queuing Telemetry Transport, 消息队列遥测传输) 是 IBM 开发的一个即时通讯协议, 它适合在低带宽、不可靠的网络中进行远程传感器和控制设备通讯等, 日益成为物联网通信协议的重要组成部分。MQTT 现在主要用于即时通讯、机器与机器的通信 (M2M) 以及物联网环境 (IoT) 等。

2.2 规格特性

MQTT 模块提供了以下功能。

表 2-1 MQTT 模块的功能特性

规格类型	支持规格	规格描述	备注
功能	消息发布	发布其它客户端可能会订阅的消息	
	订阅主题	订阅其它客户端发布的消息	
	设立遗嘱	设立遗嘱, 当断开连接后, 由服务器发布遗嘱信息	

2.3 文件位置

以 SDK 包为根目录, 本网络协议涉及到的主要文件位置如下。

表 2-2 MQTT 网络协议的文件位置

组件名	文件分类	文件位置
MQTT	源码文件	sdk/src/net/mqtt
	头文件	sdk/include/net/mqtt
	示例工程	sdk/project/example/mqtt



说明

XR806 SDK 可在以下 GitHub 仓库获取: https://github.com/XradioTech/xr806_sdk.git

3 应用说明

3.1 应用简述

MQTT 网络协议已经内嵌到 XR806 SDK 中，通过配置以及函数接口调用即可使用。

3.2 配置说明

表 3-1 XR806 MQTT 模块配置列表

配置项	配置说明
调试打印接口	<p>设置说明： 此项配置使能/关闭调试信息</p> <p>设置位置： sdk/net/mqtt/MQTTPacket/MQTTDebug.h</p> <p>设置方式：</p> <ol style="list-style-type: none"> 1. 定义宏 MQTT_DBG_ON 来启动调试，如打开调试信息： #define MQTT_DBG_ON 2. 定义宏 MQTT_ASSERT_ON 来启动断言，如打开断言信息： #define MQTT_ASSERT_ON 3. 定义宏 MQTT_PLATFORM_DBG 来启动平台移植调试信息，如打开调试信息： #define MQTT_PLATFORM_DBG

3.3 接口说明

表 3-2 MQTT 模块常用接口简介

接口名	简要介绍
初始化接口	本接口用于初始化 MQTT 环境，属于外部接口。
连接接口	本接口用于进行一次 MQTT 连接，属于外部接口。
MQTTPublish 接口	本接口用于发布消息，属于外部接口
MQTTYield 接口	本接口用于运行 MQTT 客户端，属于外部接口
MQTTSubscribe 接口	本接口用于订阅主题，属于外部接口
MQTTUnsubscribe 接口	本接口用于取消订阅主题，属于外部接口
MQTTUnsubscribe 接口	本接口用于断开连接，属于外部接口

3.3.1 初始化接口

3.3.1.1 NewNetwork

表 3-3 NewNetwork 接口函数说明

信息项	说明
原型	void NewNetwork(Network *n);
功能	初始化网络接口
参数	Network *n 含义解释：Network 结构体的指针。 使用说明：应用代码传递一个 Network 类型结构体指针即可。NewNetwork 会完成该结构体的初始化，应用代码无需了解其结构体成员
返回值	无

3.3.1.2 MQTTClient

表 3-4 MQTTClient 接口函数说明

信息项	说明
原型	void MQTTClient(Client* c, Network* network, unsigned int command_timeout_ms, unsigned char* buf, size_t buf_size, unsigned char* readbuf, size_t readbuf_size);
功能	初始化 MQTT 客户端
参数	Client* c 含义解释：Client 结构体的指针。 使用说明：应用代码传递一个 Client 类型结构体指针即可。MQTTClient 会完成该结构体的初始化，应用代码无需了解其结构体成员 Network* network 含义解释：Network 结构体的指针。 使用说明：NewNetwork 初始化好的 Network 结构体 unsigned int command_timeout_ms 含义解释：命令超时时间。 使用说明：赋值即可，单位为 ms unsigned char* buf 含义解释：发送 buffer。MQTT 借助该 buffer 进行消息发送 使用说明：赋值即可 size_t buf_size 含义解释：发送 buffer 的长度。 使用说明：赋值即可 unsigned char* readbuf 含义解释：接收 buffer。MQTT 借助该 buffer 进行消息接收。 使用说明：赋值即可

信息项	说明
	size_t readbuf_size 含义解释：接收 buffer 的长度。 使用说明：赋值即可
返回值	无

3.3.2 连接接口

3.3.2.1 ConnectNetwork

表 3-5 ConnectNetwork 接口函数说明

信息项	说明
原型	int ConnectNetwork(Network* n, char* addr, int port);
功能	以非 TLS 方式连接服务器
参数	Network* n 含义解释：Network 结构体的指针。 使用说明：NewNetwork 初始化好的 Network 结构体 char* addr 含义解释：服务器地址 使用说明：赋值即可 int port 含义解释：服务器端口号 使用说明：赋值即可
返回值	0：成功 其它：失败

3.3.2.2 TLSConnectNetwork

表 3-6 TLSConnectNetwork 接口函数说明

信息项	说明
原型	int TLSConnectNetwork(Network *n, const char *addr, const char *port, const char *ca_cert, size_t ca_cert_len, const char *client_cert, size_t client_cert_len, const char *client_key, size_t client_key_len, const char *client_pwd, size_t client_pwd_len);
功能	以 TLS 方式连接服务器
参数	Network* n 含义解释：Network 结构体的指针。 使用说明：NewNetwork 初始化好的 Network 结构体 const char *addr 含义解释：服务器地址。 使用说明：赋值即可

信息项	说明
	<p><code>const char *port</code> 含义解释：服务器端口号。 使用说明：赋值即可</p> <p><code>const char *ca_crt</code> 含义解释：CA 证书。 使用说明：赋值即可</p> <p><code>size_t ca_crt_len</code> 含义解释：CA 证书长度。 使用说明：赋值即可</p> <p><code>const char *client_crt</code> 含义解释：客户端证书。 使用说明：赋值即可</p> <p><code>size_t client_crt_len</code> 含义解释：客户端证书长度。 使用说明：赋值即可</p> <p><code>const char *client_key</code> 含义解释：客户端私钥。 使用说明：赋值即可</p> <p><code>size_t client_key_len</code> 含义解释：客户端私钥长度。 使用说明：赋值即可</p> <p><code>const char *client_pwd</code> 含义解释：客户端密码。 使用说明：赋值即可</p> <p><code>size_t client_pwd_len</code> 含义解释：客户端密码长度。 使用说明：赋值即可</p>
返回值	<p>0：成功 其它：失败</p>

3.3.2.3 MQTTConnect

表 3-7 MQTTConnect 接口函数说明

信息项	说明
原型	<code>int MQTTConnect (Client*c, MQTTPacket_connectData* connectData);</code>
功能	进行 MQTT 连接

信息项	说明
参数	<p><code>Client* c</code> 含义解释：Client 结构体的指针。 使用说明：MQTTClient 初始化好的 Client 结构体</p> <p><code>MQTTPacket_connectData* connectData</code> 含义解释：MQTTPacket_connectData 结构体指针，此结构体存放 MQTT 连接的配置信息。 使用说明：MQTTPacket_connectData 结构体的成员说明，请参阅表 3-8 MQTTPacket_connectData 结构体的成员说明。</p>
返回值	<p>0：成功 其它：失败</p>

表 3-8 MQTTPacket_connectData 结构体的成员说明

成员项	说明
<code>char struct_id[4];</code>	<p>含义解释：MQTT 识别码。 使用说明：必须设置为 MQTT</p>
<code>int struct_version;</code>	<p>含义解释：结构体的版本号。 使用说明：赋值为 0</p>
<code>unsigned char MQTTVersion;</code>	<p>含义解释：mqtt 的版本。 使用说明：赋值为 3，则版本为 3.1；赋值为 4，则版本为 3.1.1</p>
<code>MQTTString clientId;</code>	<p>含义解释：客户端的 id 号。 使用说明：赋值即可</p>
<code>unsigned short keepAliveInterval;</code>	<p>含义解释：心跳时间。 使用说明：赋值即可</p>
<code>unsigned char cleansession;</code>	<p>含义解释：是否使能服务器的 CleanSession 选项。 使用说明：0：不使能；1：使能</p>
<code>unsigned char willFlag;</code>	<p>含义解释：遗嘱设置标志位。 使用说明：0：无遗嘱，1：有遗嘱</p>
<code>MQTTPacket_willOptions will;</code>	<p>含义解释：遗嘱消息。 使用说明：赋值即可</p>
<code>MQTTString username;</code>	<p>含义解释：连接时的用户名。 使用说明：赋值即可</p>

成员项	说明
MQTTString password;	含义解释：连接时的密码。 使用说明：赋值即可

3.3.3 MQTTPublish 接口

3.3.3.1 MQTTPublish

表 3-9 MQTTPublish 接口函数说明

信息项	说明
原型	int MQTTPublish (Client* c, const char* topicName, MQTTMessage* message);
功能	发布一个消息
参数	Client* c 含义解释：Client 结构体的指针。 使用说明：MQTTClient 初始化好的 Client 结构体 const char* topicName 含义解释：发布消息的主题。 使用说明：赋值即可。 MQTTMessage* message 含义解释：发布的消息。 使用说明：赋值即可。
返回值	0：成功 其它：失败

3.3.4 MQTTYield 接口

3.3.4.1 MQTTYield

表 3-10 MQTTYield 接口函数说明

信息项	说明
原型	int MQTTYield (Client* c, int timeout_ms);
功能	MQTT 帧接收、处理、发送心跳包等。该接口需要不断被调用，且该函数是阻塞式的，在 timeout 前，会阻塞同优先级的其他线程，导致其他线程不能正常运行，可根据要求自行调整 mqtt 线程优先级、减小 timeout 时间、提高调用 MQTTYield 的频率（可用系统延时），保证在每次调用的时间间隔内有时间执行其他操作，从而做到尽量减少对线程的阻塞。
参数	Client* c 含义解释：Client 结构体的指针。 使用说明：MQTTClient 初始化好的 Client 结构体 int timeout_ms

信息项	说明
	含义解释：阻塞时间。 使用说明：赋值即可。
返回值	0：失败 1：成功

3.3.5 MQTTSubscribe 接口

3.3.5.1 MQTTSubscribe

表 3-11 MQTTSubscribe 接口函数说明

信息项	说明
原型	int MQTTSubscribe (Client* c, const char* topicFilter, enum QoS qos, messageHandler messageHandler);
功能	向服务器订阅主题
参数	<p>Client* c 含义解释：Client 结构体的指针。 使用说明：MQTTClient 初始化好的 Client 结构体</p> <p>const char* topicFilter 含义解释：订阅的主题。 使用说明：赋值即可</p> <p>enum QoS qos 含义解释：订阅的服务质量等级。 使用说明：赋值即可</p> <p>messageHandler messageHandler 含义解释：回调函数。接收到该订阅的主题时，触发的回调处理函数 使用说明：赋值即可</p>
返回值	0：失败 1：成功

3.3.6 MQTTUnsubscribe 接口

3.3.6.1 MQTTUnsubscribe

表 3-12 MQTTUnsubscribe 接口函数说明

信息项	说明
原型	int MQTTUnsubscribe (Client* c, const char* topicFilter);
功能	取消订阅主题
参数	<p>Client* c 含义解释：Client 结构体的指针。 使用说明：MQTTClient 初始化好的 Client 结构体</p>

信息项	说明
	const char* topicFilter 含义解释：取消订阅的主题。 使用说明：赋值即可
返回值	0：失败 1：成功

3.3.7 MQTTUnsubscribe 接口

3.3.7.1 MQTTDisconnect

表 3-13 MQTTDisconnect 接口函数说明

信息项	说明
原型	int MQTTDisconnect (Client* c);
功能	断开连接
参数	Client* c 含义解释：Client 结构体的指针。 使用说明：MQTTClient 初始化好的 Client 结构体
返回值	0：失败 1：成功

4 示例说明

MQTT 接口用于进行 MQTT 连接通信，下面以 MQTT 连接通信为例，简要说明 MQTT 接口的使用。

4.1 示例简介

MQTT 示例的演示的目的是简要介绍 MQTT 接口的基本使用方法，演示的功能为连接服务器，然后进行主题订阅、主题发布等。

4.1.1 获取方式

MQTT 示例有示例工程代码，位于 XR806 SDK 的 /project/example/mqtt 目录，以下此示例工程简称为 MQTT 示例工程。



说明

XR806 SDK 可在以下 GitHub 仓库获取：https://github.com/XradioTech/xr806_sdk.git

4.1.2 准备工作

MQTT 示例工程的硬件准备如下。

1. 评估板：运行示例工程代码。
2. 串口线：连接评估板的 Uart0 插针，用于 console 控制台的输入输出。
3. PC 机：用于镜像烧录和 console 控制的输入输出。

MQTT 的软件准备，包括烧写工具、代码编译和烧写操作，请参见《XR806_SDK_快速入门指南》

4.1.3 操作步骤

开发板完成烧写，复位，通过串口命令连接网络后，示例代码自动运行。

表 4-1 MQTT 示例工程的命令列表

命令	说明
快速配置网络	<p>命令格式： net sta config <ssid> [psk]</p> <p>命令解释： 快速配置被连接 AP 的 SSID 和密码</p> <p>参数说明： ssid：被连接 AP 的 ssid psk：连接 AP 的密码，如果为开放网络，则不需要此项。</p>
启用网络	<p>命令格式： net sta enable</p>

命令	说明
	命令解释： 开始连接 AP，需要先完成网络配置

4.2 效果展示

在评估板中运行 MQTT 示例程序后，在控制台中会打印出 MQTT 的连接情况，并显示消息主题接收过程，如下所示。

```
mqtt connected
get a message, topic: /vo84Hm3xbUj/xr806_senor_set0/data, msg: mqtt demo test
get a message, topic: /vo84Hm3xbUj/xr806_senor_set0/data, msg: mqtt demo test
get a message, topic: /vo84Hm3xbUj/xr806_senor_set0/data, msg: mqtt demo test
get a message, topic: /vo84Hm3xbUj/xr806_senor_set0/data, msg: mqtt demo test
```

4.3 实现流程

第一步：初始化网络实体、客户端实体

```
Client client;
Network network;
unsigned char sendbuf[80], readbuf[80];

NewNetwork(&network);
MQTTClient(&client, &network, 30000, sendbuf, sizeof(sendbuf), readbuf, sizeof(readbuf));
```

第二步：连接服务器。如连接 iot.eclipse.org 服务器，端口号 1883

```
char* address = "iot.eclipse.org";
rc = ConnectNetwork(&network, address, 1883);
if (rc != 0) {
    printf("Return code from network connect is %d\n", rc);
    return -1;
}
```

如果使用 TLS 加密，则使用 TLSConnectNetwork：

```
char* address = "iot.eclipse.org";
rc = TLSConnectNetwork(&network, address, 1883, NULL, 0, NULL, 0, NULL, 0, NULL, 0);
if (rc != 0) {
    printf("Return code from network connect is %d\n", rc);
    return -1;
}
```

第三步：连接 MQTT 服务器

```
MQTTPacket_connectData connectData = MQTTPacket_connectData_initializer;
connectData.MQTTVersion = 4;
connectData.clientID.cstring = "XR_Radio_test0";
```

```
if ((rc = MQTTConnect(&client, &connectData)) != 0)
    printf("Return code from MQTT connect is %d\n", rc);
```

第四步：订阅主题（可选）

(1) 首先实现一个回调函数，用于处理在该主题中收到的消息

```
void messageArrived(MessageData* data)
{
    printf("Message arrived on topic %.*s: %.*s\n", data->topicName->lenstring.len,
    data->topicName->lenstring.data, data->message->payloadlen, (char *)data->message->payload);
}
```

(2) 订阅主题

```
if ((rc = MQTTSubscribe(&client, "XR_Radio/test/#", QOS1, messageArrived)) != 0)
    printf("Return code from MQTT subscribe is %d\n", rc);
```

第五步：发布消息（可选）

```
MQTTMessage message;
char payload[30] = "message number 0";

message.qos = QOS1;
message.retained = 0;
message.payload = payload;
message.payloadlen = strlen(payload) + 1;

if ((rc = MQTTPublish(&client, "XR_Radio/test/a", &message)) != 0)
    printf("Return code from MQTT publish is %d\n", rc);
```

第六步：取消订阅（可选）

```
if ((rc = MQTTUnsubscribe(&client, "XR_Radio/test/#")) != 0)
    printf("Return code from MQTT unsubscribe is %d\n", rc);
```

第七步：断开连接

```
if (client.isconnected) {
    if ((rc = MQTTDisconnect(&client)) != 0) {
        printf("Return code from MQTT disconnect is %d\n", rc);
        network.disconnect(&network);
        return -1;
    }
    network.disconnect(&network);
}
```

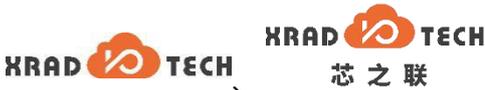
著作权声明

版权所有©2020 广州芯之联科技有限公司。保留一切权利。

本档及内容受著作权法保护，其著作权由广州芯之联科技有限公司（“芯之联”）拥有并保留一切权利。

本档是芯之联的原创作品和版权财产，未经芯之联书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本档内容的部分或全部，且不得以任何形式传播。

商标声明



KRAD TECH、**芯之联**（不完全列举）均为广州芯之联科技有限公司的商标或者注册商标。在本档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与广州芯之联科技有限公司（“芯之联”）之间签署的商业合同和条款的约束。本档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，芯之联概不负责。

本档作为使用指导仅供参考。由于产品版本升级或其他原因，本档内容有可能修改，如有变更，恕不另行通知。芯之联尽全力在本档中提供准确的信息，但并不确保内容完全没有错误，因使用本档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，芯之联概不负责。本档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本档未以明示或暗示或其他方式授予芯之联的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。芯之联不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。芯之联不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。