

ESP8266 LUA开发基础入门篇
备份(22)
ESP8266 SDK开发(32)
ESP8266 SDK开发基础入门篇
备份(30)
GPRS Air202 LUA开发(11)
HC32F460(华大) +
BC260Y(NB-IOT) 物联网开发
(5)
NB-IOT Air302 AT指令和LUA
脚本语言开发(25)
PLC(三菱PLC)基础入门篇(2)
STM32+Air724UG(4G模组)
物联网开发(43)
STM32+BC26/260Y物联网开
发(37)
STM32+CH395Q(以太网)物
联网开发(7)
STM32+ESP8266(ZLESP8266/
物联网开发(1)
STM32+ESP8266+AIR202/30:
远程升级方案(16)
STM32+ESP8266+AIR202/30:
终端管理方案(6)
STM32+ESP8266+Air302物
联网开发(58)
STM32+W5500+AIR202/302
基本控制方案(25)
STM32+W5500+AIR202/302
远程升级方案(6)
UCOSii操作系统(1)
W5500 学习开发(8)
编程语言C#(11)
编程语言Lua脚本语言基础入
门篇(6)
编程语言Python(1)
单片机(LPC1778)LPC1778(2)
单片机(MSP430)开发基础入门
篇(4)
单片机(STC89C51)单片机开发
板学习入门篇(3)
单片机(STM32)基础入门篇(3)
单片机(STM32)综合应用系列
(16)
电路模块使用说明(10)
感想(6)
软件安装使用: MQTT(8)
软件安装使用: OpenResty(6)
更多

最新评论

1. Re:C#开发: 通信篇-TCP客
户端
感谢分享, 直接就用上了
--Zfen
2. Re:03-STM32+Air724UG
远程升级篇OTA(阿里云物联
网平台)-STM32+Air724UG
使用阿里云物联网平台OTA
远程更新STM32程序
楼主, 单片机和Air724模块
之间是通过AT指令通讯的
吗?
--a314825348



阅读排行榜

1. ESP8266使用详解(AT,LUA,
SDK)(172474)
2. 1-安装MQTT服务器(Windo
ws),并连接测试(97912)

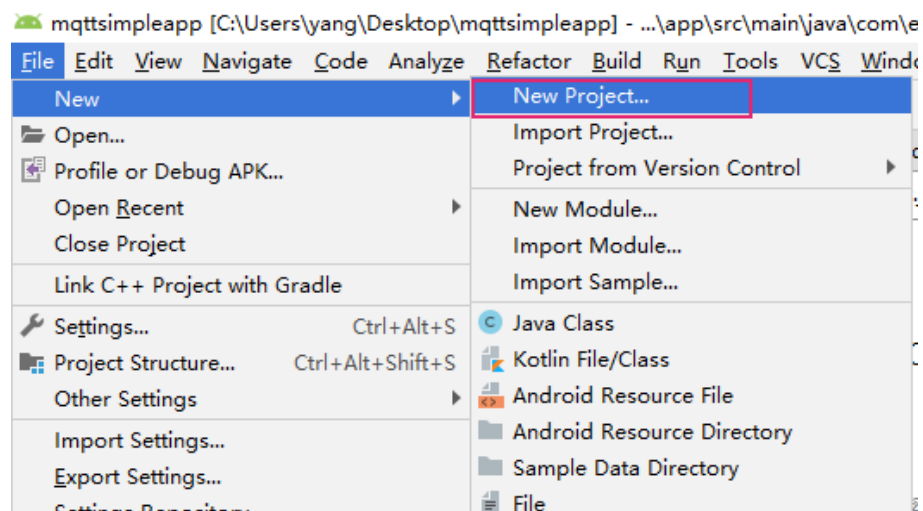
说明

这一节编写android连接MQTT服务器程序

提示:为了便于移植扩展使用,我封装了下MQTT.这节只说明在
MyMqttClint.java的移植使用

 MyMqttClient.java
 org.eclipse.paho.client.mqttv3-1.2.0.jar

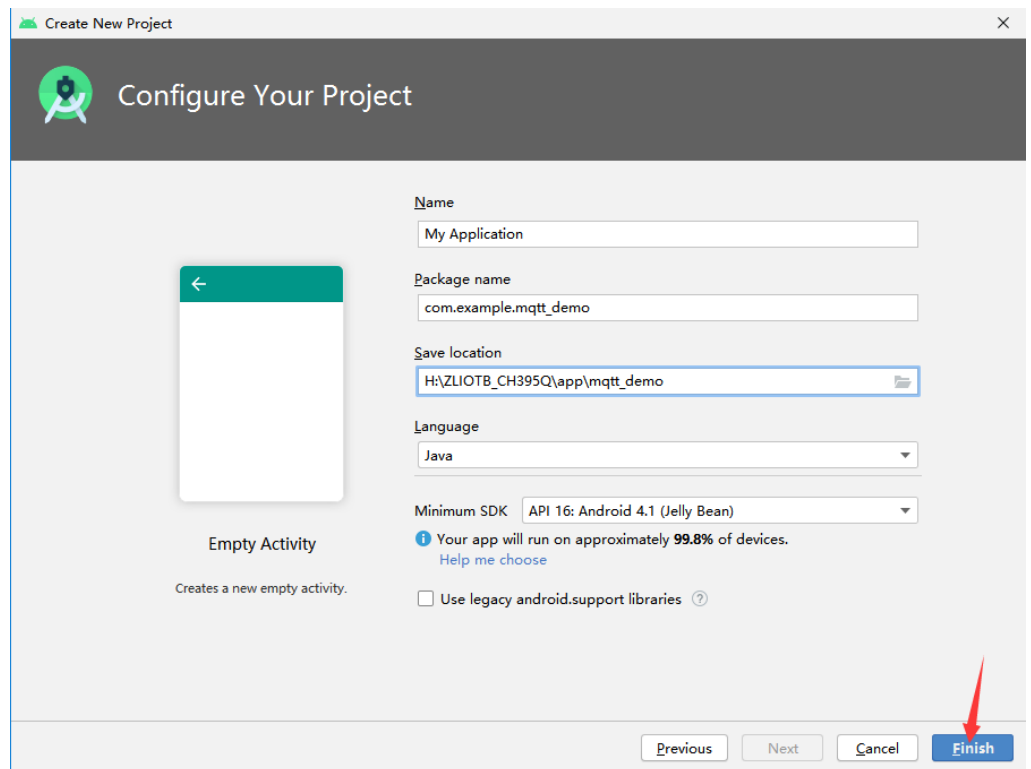
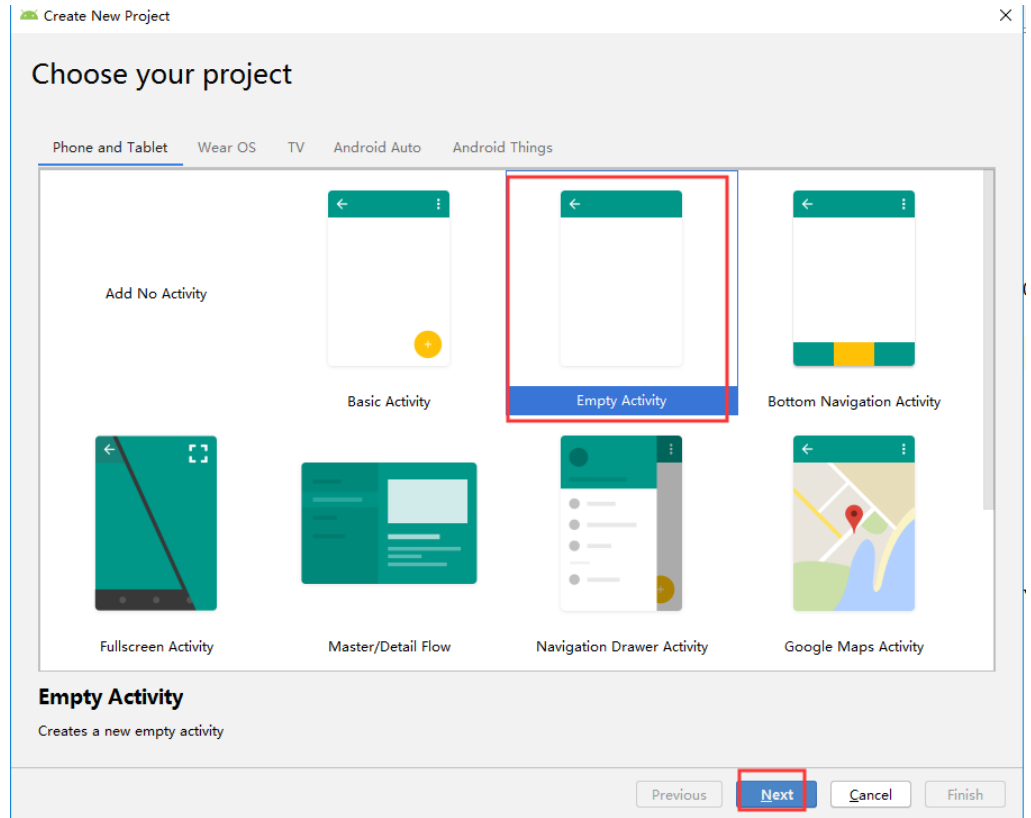
新建工程



3. ESP8266刷AT固件与node mcu固件(64263)
4. 用ESP8266+android,制作自己的WIFI小车(ESP8266篇)(63484)
5. 有人WIFI模块使用详解(38337)
6. (一)基于阿里云的MQTT远程控制(Android 连接MQTT服务器,ESP8266连接MQTT服务器实现远程通信控制----简单的连接通信)(35733)
7. 关于TCP和MQTT之间的转换(32824)
8. C#中public与private与static(31833)
9. android 之TCP客户端编程(31657)
10. android客服端+eps8266+单片机+路由器之远程控制系统(31244)

推荐排行榜

1. C#委托+回调详解(9)
2. 用ESP8266+android,制作自己的WIFI小车(ESP8266篇)(8)
3. 用ESP8266+android,制作自己的WIFI小车(Android 软件)(6)
4. ESP8266使用详解(AT,LUA,SDK)(6)
5. 关于TCP和MQTT之间的转换(5)

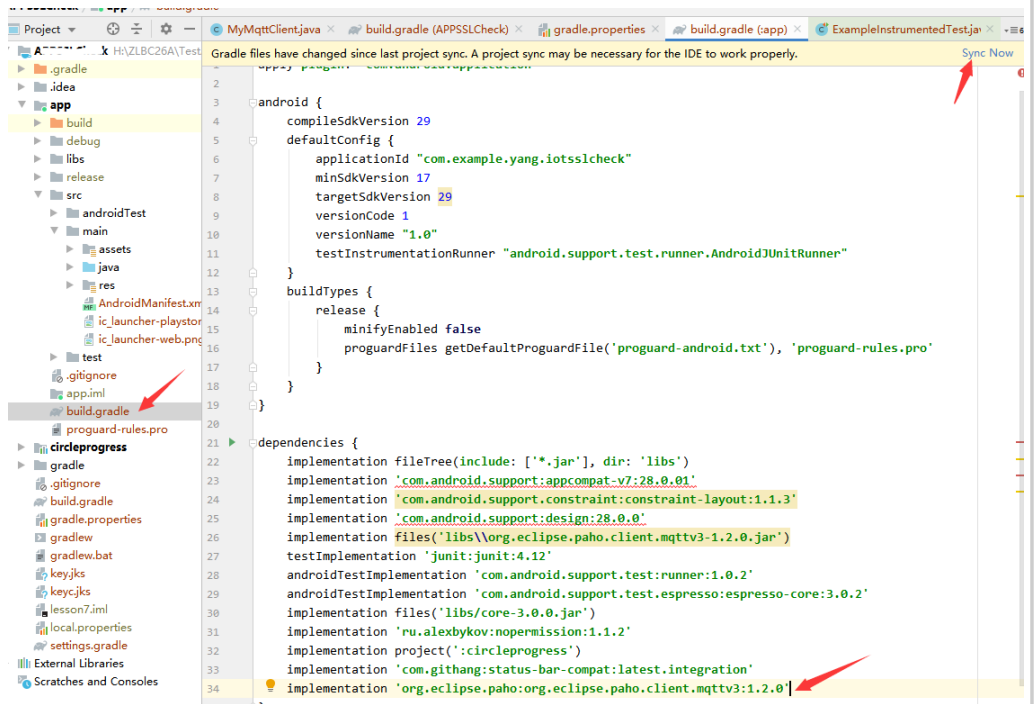


安装MQTT的Jar包

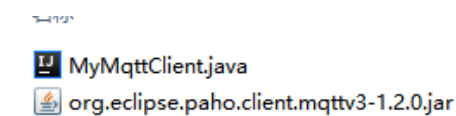
1.第一种方式

implementation

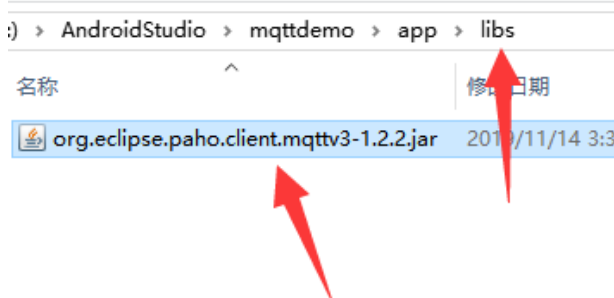
'org.eclipse.paho:org.eclipse.paho.client.mqttv3:1.2.0'

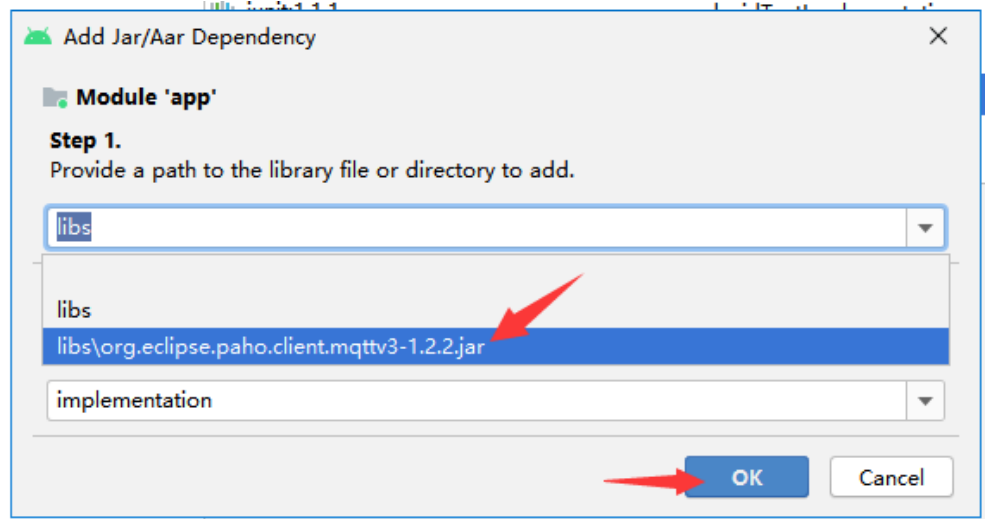
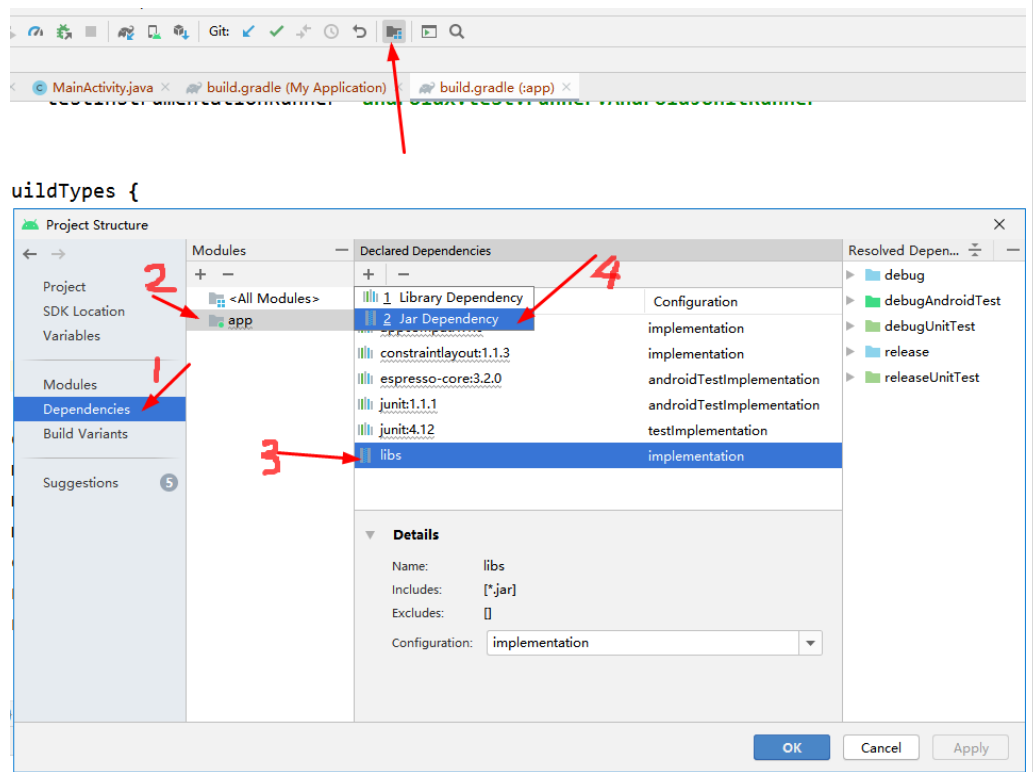


2.第二种,我准备好了jar包,可以直接导入 jar包

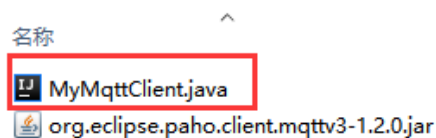


拷贝到自己工程的libs里面





把MyMqttClient.java 文件放到自己的工程



1.根据自己的MQTT服务器修改MQTT信息

```
1 package com.example.myapplicationmqttdemo;
2
3 import ...
4
5 public class MyMqttClient {
6     private static final String TAG = MyMqttClient.class.getSimpleName();
7     private static MyMqttClient myMqttClient;
8     public static String clientId = ""; //clientId, 如果不设置, 默认随机数
9     public static String mqttUserString = "yang"; //用户名
10    public static String mqttPwdString = "11223344"; //密码
11    public static String mqttIPString = "mniif.cn"; //IP地址
12    public static int mqttPort = 1883; //端口号
13    public static int KeepAlive = 60; //心跳包时间
14    public static boolean useSSL = false; //是否使用SSL; 默认单向认证, 忽略证书连接.
15    public static boolean CleanSession = true; //清空session
16
17
18    private static MqttClient mqttClient;
19    private static MqttConnectOptions mqttConnectOptions;
20    private boolean ConnectFlage = true;
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
```

2,增加网络权限

```
<uses-permission
android:name="android.permission.INTERNET" />
```

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.myapplicationmqttdemo">
4
5     <uses-permission android:name="android.permission.INTERNET" />
6
7     <application
8         android:allowBackup="true"
9         android:icon="@mipmap/ic_launcher"
10        android:label="My Application"
11        android:roundIcon="@mipmap/ic_launcher_round"
12        android:supportRtl="true"
13        android:theme="@style/AppTheme">
14        <activity android:name=".MainActivity">
15            <intent-filter>
16                <action android:name="android.intent.action.MAIN" />
```

连接MQTT

1.把以下代码在初始化的时候调用一次,该代码写一次即可,后期断线自动重连

```
MyMqttClient.sharedCenter().setConnect();
```

```
package com.example.myapplicationmqttdemo;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        MyMqttClient.sharedCenter().setConnect();//连接MQTT(自动掉线重连)
    }
}
```

2.安装运行APP

```
Logcat
Vivo Vivo X21UD A Android 9, A com.example.mqtt_demo (10779) Error
logcat
2021-07-15 12:20:03.089 10779-10779/com.example.mqtt_demo E/MyMqttClient: sharedCenter: myMqttClient
2021-07-15 12:20:03.507 10779-11124/com.example.mqtt_demo E/MyMqttClient: run: Connect Success
```

订阅主题

MyMqttClient.sharedCenter().setSubscribe("1111",0);//订阅主题1111,消息等级0

```
package com.example.myapplicationmqttdemo;

import ...

public class MainActivity extends AppCompatActivity {

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        MyMqttClient.sharedCenter().setConnect();//连接MQTT(自动掉线重连)
        //订阅主题1111,消息等级0
        MyMqttClient.sharedCenter().setSubscribe( Topic "1111", qos: 0);
    }
}
```

上面这样子做有可能订阅不成功,因为可能还没连接上,

建议使用下面的方式, 设置订阅成功回调,用一个定时器持续订阅



```
public class MainActivity extends AppCompatActivity {
    //定时器用于轮训订阅主题
    private Timer timerSubscribeTopic = null;
    private TimerTask TimerTaskSubscribeTopic = null;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);

        MyMqttClient.sharedCenter().setConnect();

        //MQTT连接成功回调
        MyMqttClient.sharedCenter().setOnServerConnectedCallback(new MyMqttCl
            @Override
            public void callback() {
                startTimerSubscribeTopic();//定时订阅主题
            }
        });

        //订阅主题成功回调
        MyMqttClient.sharedCenter().setOnServerSubscribeCallback(new MyMqttCl
            @Override
            public void callback(String Topic, int qos) {
                if (Topic.equals("1111")){//订阅1111成功
                    stopTimerSubscribeTopic();//订阅到主题,停止订阅
                }
            }
        });
        startTimerSubscribeTopic();//定时订阅主题
    }

    /**
     * 定时器每隔1s尝试订阅主题
     */
    private void startTimerSubscribeTopic(){
        if (timerSubscribeTopic == null) {
            timerSubscribeTopic = new Timer();
        }
        if (TimerTaskSubscribeTopic == null) {
            TimerTaskSubscribeTopic = new TimerTask() {
                @Override
                public void run() {
                    MyMqttClient.sharedCenter().setSubscribe("1111",0);//订阅
                }
            };
        }
        if(timerSubscribeTopic != null && TimerTaskSubscribeTopic != null )
            timerSubscribeTopic.schedule(TimerTaskSubscribeTopic, 0, 1000);
    }

    private void stopTimerSubscribeTopic(){
```

```

        if (timerSubscribeTopic != null) {
            timerSubscribeTopic.cancel();
            timerSubscribeTopic = null;
        }
        if (TimerTaskSubscribeTopic != null) {
            TimerTaskSubscribeTopic.cancel();
            TimerTaskSubscribeTopic = null;
        }
    }

    //当活动不再可见时调用
    @Override
    protected void onStop()
    {
        super.onStop();
        stopTimerSubscribeTopic(); //停止定时器订阅
    }

    /**
     * 当处于停止状态的活动需要再次展现给用户的时候，触发该方法
     */
    @Override
    protected void onRestart() {
        super.onRestart();
        startTimerSubscribeTopic(); //定时订阅主题
    }

    @Override
    protected void onPause() {
        super.onPause();
        stopTimerSubscribeTopic();
    }
}

```

注意:这两个地方不可省略,只要进入页面就启动定时器订阅,如果在运行过程中MQTT断线重连了,则接着重新订阅

```

activity_main.xml x MainActivity.java x MyMqttClient.java x AndroidManifest.xml x
11 //定时器用于轮训订阅主题
12 private Timer timerSubscribeTopic = null;
13 private TimerTask TimerTaskSubscribeTopic = null;
14 @Override
15 protected void onCreate(Bundle savedInstanceState) {
16     super.onCreate(savedInstanceState);
17     setContentView(R.layout.activity_main);
18
19     MyMqttClient.sharedCenter().setConnect();
20
21     //MQTT连接成功回调
22     MyMqttClient.sharedCenter().setOnServerConnectedCallback(new MyMqttClient.OnServerConnectedCallback() {
23         @Override
24         public void callback() {
25             startTimerSubscribeTopic(); //定时订阅主题
26         }
27     });
28
29     //订阅主题成功回调
30     MyMqttClient.sharedCenter().setOnServerSubscribeCallback(new MyMqttClient.OnServerSubscribeSuccessCallback() {
31         @Override
32         public void callback(String Topic, int qos) {
33             if (Topic.equals("1111")){//订阅1111成功
34                 stopTimerSubscribeTopic(); //订阅到主题, 停止订阅
35             }
36         }
37     });
38     startTimerSubscribeTopic(); //定时订阅主题
39 }
40
41 /**

```


取消订阅

```
MyMqttClient.sharedCenter().setUnSubscribe("1111");//取消  
订阅主题1111
```

接收消息

哪个文件需要获取MQTT数据,只需要按照下面的方式写上即可!

```
MyMqttClient.sharedCenter().setOnServerReadStringCallback(new MyMqttClient.On  
    @Override//Topic:主题 Msg.toString():接收的消息 MsgByte:16进制消息  
    public void callback(String Topic, MqttMessage Msg, byte[] MsgByte  
        Log.e("MqttMsg", "Topic: "+Topic+" Msg"+Msg.toString() );  
    }  
});
```

发布消息

1.发布字符串消息

```
MyMqttClient.sharedCenter().setSendData("2222","msg",0,fa  
lse);//发送字符串消息
```

2222:发布的主题 msg:发布的消息 0:消息等级 false:不需要服务器记录

2.发布16进制消息

```
byte[] bytes= new byte[2];
```

```
bytes[0] = 0x03;
```

```
bytes[1] = 0x55;
```

```
MyMqttClient.sharedCenter().setSendData("3333",bytes,0,fa  
lse);
```

监听MQTT连接状态

```
/*连接上MQTT*/  
MyMqttClient.sharedCenter().setOnServerConnectedCallback(new MyMqttCL  
    @Override  
    public void callback() {
```

```
    }
  });
  /*和MQTT服务器断开连接*/
  MyMqttClient.sharedCenter().setOnServerDisConnectedCallback(new MyMqt
    @Override
    public void callback(Throwable e) {

    }
  });
});
```

分类: [STM32+CH395Q\(以太网\)物联网开发](#)

好文要顶

关注我

收藏该文



杨奉武

关注 - 1

粉丝 - 621

0

0

« 上一篇: [2-STM32+CH395Q\(以太网\)基本控制篇\(自建物联网平台\)-移植单片机MQTT包,编写stm32+CH395Q连接MQTT服务器程序](#)

posted on 2021-07-15 12:32 杨奉武 阅读(0) 评论(0) 编辑 收藏 举报

[刷新评论](#) [刷新页面](#) [返回顶部](#)

发表评论

[编辑](#) [预览](#)

B

支持 Markdown

自动补全

[提交评论](#) [退出](#)

[Ctrl+Enter快捷键提交]

【推荐】 百度智能云特惠: 新用户首购云服务器低至0.7折, 个人企业同享

【推荐】 大型组态、工控、仿真、CAD\GIS 50万行VC++源码免费下载!

【推荐】 阿里云云大使特惠: 新用户购ECS服务器1核2G最低价87元/年

【推荐】 投资训练营: 一杯咖啡的价格, 教你学会投资, 增加被动收入

【推荐】 加州大学伯克利分校高管教育: 大数据与数学科学-在线课程

【推荐】 和开发者在一起: 华为开发者社区, 入驻博客园科技品牌专区