

# ICP DAS SmartQ SDK User Manual

[Version 1.0]



**ICP DAS CO., LTD.**

泓格科技股份有限公司

---

## 免责声明 **Warning**

泓格科技股份有限公司对于因为应用本产品所造成的损害并不负任何法律上的责任。本公司保留有任何时间未经通知即可变更与修改本档内容之权利。本文所含信息如有变更，恕不予另行通知。

本公司尽可能地提供正确与可靠的信息，但不保证此信息的使用或其它团体在违反专利或权利下使用。此处包涵的技术或编辑错误、遗漏，概不负其法律责任。

## 版权 **Copyright**

© 2009 泓格科技股份有限公司保留所有权利。

## 商标识别 **Trademark**

本档提到的所有公司商标、商标名称及产品名称分别属于该商标或名称的拥有者所有。

## 授权宣告 **License**

使用者仅被授权可以在单一计算机上与有限条件下使用、备份软件与相关数据，不得同时于该单一计算机外使用本软件。本公司仍保有此软件与相关数据的著作权及其它智能财产权。除非事先经过本公司的书面授权，否则禁止重制、传送及散布等方式取得部份或全部软件或相关的复制品。

## 目录：

1. 建置环境 .....	4
1.1 C / C++ .....	4
1.2 C# / VB .Net .....	5
2. 函式界面 .....	7
2.1 自定义结构 PPARMS:.....	7
2.2 函式： .....	10
3. 范例程序说明 .....	16
3.1 Publish.c (Publish_WinCE.cpp) .....	16
3.2 SubScribe.c (SubScribe _WinCE.cpp) .....	18

## 1. 建置环境

在建置程序之前，建议使用者先参考WinPAC-8000 User Manual ([WinPAC 8x4x user manual v2.0.1.pdf](#))，初步了解整个开发环境架设与开发步骤。

手册下载网址：

[http://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x\\_ce50/document/winpac\\_8x4x\\_user\\_manual\\_v2.0.1.pdf](http://ftp.icpdas.com/pub/cd/winpac/napdos/wp-8x4x_ce50/document/winpac_8x4x_user_manual_v2.0.1.pdf)

### 1.1 C / C++

#### ● Windows

1. 建立一个Visual Studio C++的Win32 Console空项目。
2. 将Publish.c、SmartQAPI.h、MQTT.dll及MQTT.lib等4个档案复制到该项目档夹内。
3. 将SmartQAPI.h及Publish.c汇入该专案。
4. 于该项目的连接器中加入ws2\_32.lib及MQTT.lib。
5. 将Publish.c的Precompiled Headers设定为Not using precompiled headers。
6. 重复1-5步骤，唯其中2、3、5步骤中Publish.c档案改为SubScribe.c档案。
7. 执行Publish及SubScribe程序。

#### ● QP-8x1/QP-8x2

1. 选定已安装eMbedded Visual C++的计算机为程序开发平台，您也可以至Microsoft网站免费下载eMbedded Visual C++。
2. 安装WinPAC SDK：pac270\_sdk\_20091006.msi (已放置于 SmartQ CD-ROM：\SmartQ SDK\ PAC270 SDK 目录中)或以后的版本于此平台。(请参考WinPAC-8000 User Manual 第4.2节 “Installing WinPAC SDKs” )
3. 建立一个eMbedded Visual C++的WCE Application项目。(请参考WinPAC-8000 User Manual 第4.6节 “Your First Program in eMbedded Visual C++”)
4. 将Publish\_WinCE.cpp、SmartQAPI.h、WMQTT\_CE\_MT.dll、WMQTT\_CE\_MT.lib等4个档案复制到该项目档夹内。
5. 将SmartQAPI.h及Publish\_WinCE.cpp汇入该专案。
6. 于该项目的连接器中加入WMQTT\_CE\_MT.lib。
7. 将Publish\_WinCE.cpp的Precompiled Headers设定为Not using precompiled headers。

8. 于该项目的WCE Configuration工具列(tool bar)中将STANDARDSDK改为PAC270。
9. 并于主选单Tools\Configure Platform Manager中选取PAC270 Device, 进入Device Properties页面将Transport设定为TCP/IP Transport for Windows CE, 以及Startup Server设定为Manual Server。
10. 执行程序, 出现CEMGRC.EXE的cmd指令。
11. 将CEMGRC.EXE的cmd指令输入QP-8x1/QP-8x2 WinCE中。
12. 执行QP-8x1/QP-8x2 WinCE内下载回来的执行文件程序(默认储存于Micro\_SD档夹内, 另外WMQTT\_CE\_MT.dll必须放入执行文件的目录当中)。
13. 建立SubScribe项目: 重复3-12步骤, 唯其中4、5、7步骤中Publish\_WinCE.cpp档案改为SubScribe\_WinCE.cpp档案。
14. 执行Publish及SubScribe程序。

## 1.2 C# / VB .Net

### ● Windows

1. 建立一个Visual Studio C#(或VB)的Console应用程序(Console Application)空项目。
2. 将Publish.cs(或Publish.vb)、MQTT.dll与MQTTDotNet.dll等3个档案到该项目档案夹内。
3. 在方案总管” Solution Explorer” 窗口中, 右键点选参考Reference, 或点选项目属性(properties)选择参考页签, 新增MQTTDotNet.dll的参考路径。
4. 将Publish.cs(或Publish.vb)汇入该项目, 确认档案开头有加入using ICPDAS.SmartQ(或Import ICPDAS.SmartQ)。
5. 重复1-4步骤, 唯其中2、4步骤中Publish.cs(或Publish.vb)档案改为SubScribe.cs(或SubScribe.vb)档案。
6. 另外MQTT.dll与MOTTDotNet.dll必须放入执行文件的目录当中。
7. 执行Publish及SubScribe程序。

### ● QP-8x1/QP-8x2

1. 安装WinPAC SDK: pac270\_sdk\_20091006.msi (已放置于 SmartQ CD-ROM : \SmartQ SDK\ PAC270 SDK 目录中)或以后的版本于此平台。(请参考WinPAC-8000 User Manual 第4.2节 “Installing WinPAC SDKs” )
2. 建立一个Visual Studio C#(或VB)的Smart Device中Windows CE 5.0

的Console应用程序(Console Application)空项目。(请参考 WinPAC-8000 User Manual 第4.4节 “Your First Program in C#” 或第4.5节 “Your First Program in VB.net”)

3. 将Publish.cs(或Publish.vb)、WMQTTDotNetCF.dll、WMQTT\_CE\_MT.dll等3个档案复制到该项目档案夹内。
4. 在方案总管” Solution Explorer” 窗口中, 在项目节点上按右键选择参考Reference, 或点选项目属性(properties)选择参考页签, 新增WMQTTDotNetCF.dll的参考路径。
5. 在项目节点上按右键选择Change Target Platform, 在Change To选项中, 将平台改为PAC270。
6. 在设备(Device)页签中, 将Target Device改选为” PAC50-ARMV4:ARMV4I\_Releas”
7. 将Publish.cs(或Publish.vb)汇入该项目, 确认档案开头有using ICPDAS.SmartQ(或Import ICPDAS.SmartQ)。
8. 建立SubScribe项目: 重复2-7步骤, 唯其中3、7步骤中Publish.cs档案改为SubScribe.cs(或SubScribe.vb)档案。
9. 参考QPAC硬件使用手册, 编译完后将程序以FTP上传到QPAC的存放目录。
10. **WMQTTDotNetCF.dll、WMQTT\_CE\_MT.dll必须放入执行文件的目录当中**
11. 执行Publish及SubScribe程序。

## 2. 函式界面

### 2.1 自定义结构 PSPARMS:

参数名称	类型	用途
pBroker	char* / String	用以储存SmartQ QP-500的网络地址或主机名称。
Port	Int	SmartQ QP-8x1/QP-8x2 or Windows Client与SmartQ QP-500连接的通讯端口号码，标准设置为1883。
Topic	char* / String	<p>SmartQ QP-8x1/QP-8x2 or Windows Client发布至SmartQ QP-500的讯息主题或及SmartQ QP-8x1/QP-8x2 or Windows Client与SmartQ QP-500订阅的讯息主题。一般而言，讯息主题常见三种特殊字符：</p> <p>分隔符： (/)</p> <p>单一层级： (+)</p> <p>多层层级： (#)</p> <p>举例说明：</p> <p>例1: topic = LEVEL1     讯息主题为LEVEL1</p> <p>例2: topic = LEVEL1/LEVEL2     讯息主题为LEVEL1/LEVEL2，分隔符(/)在此可视为LEVEL2继承LEVEL1主题，而非LEVEL1或LEVEL2主题。</p> <p>例3: topic = LEVEL1/LEVEL2/+     对于LEVEL1/LEVEL2/及所有LEVEL1/LEVEL2/LEVEL3共三层的主题感兴趣。</p> <p>例4: topic = LEVEL1/LEVEL2/#     对于所有继承LEVEL1/LEVEL2的主题全部感兴趣。</p>
qos	Int	用以设定SmartQ QP-8x1/QP-8x2 or Windows Client与SmartQ QP-500间讯息传输的服务质量，Qos值越高则传输质量越高，相对地网络负荷则越重。其值可

		<p>为0,1或2，默认值为0。</p> <p>当Qos值为：</p> <p>0：最多仅传送一次，不论传送是否成功皆不再传送。</p> <p>1：至少传送一次，直到确保SmartQ QP-500成功接收。</p> <p>2：准确地传送一次，重复发送直到确认接收者接收到为止。</p> <p><b>建议连接同一个SmartQ QP-500的所有SmartQ QP-8x1/QP-8x2/QP-711/QP-712 or Windows Client，都使用相同的QOS值，以免发生不可预测的错误。</b></p>
retain	Int	Publish所传输的讯息是否被QP-500所保留，默认值为0(不保留)。
timeout	Int	程序接收讯息时的等待时间，单位为秒，默认值为10(秒)。
keepAliveTime	Unsigned short / ushort	SmartQ QP-8x1/QP-8x2 or Windows Client与SmartQ QP-500之间联机的存活时间。当SmartQ QP-8x1/QP-8x2 or Windows Client在此设定时间内都没有发送讯息给SmartQ QP-500时，此联机就会被SmartQ QP-500强制中断。
hConn	void* / IntPtr	用以纪录有效联机的内存地址。
lastSentMsg	Long / ulong	用以纪录处理的状况。
clientId	char* / String	SmartQ QP-8x1/QP-8x2 or Windows Client用以连接至SmartQ QP-500的使用者名称。不可以两个以上的client同时有相同名称。
lwtTopic	char* / String	离线主题，设定SmartQ QP-8x1/QP-8x2 or Windows Client离线后SmartQ QP-500会自动发送的讯息主题。
lwtData	char* / String	离线数据，设定SmartQ QP-8x1/QP-8x2 or Windows Client离线后SmartQ QP-500会自动发送的讯息数据。
lwtQos	int	遗言的QOS设定。



lwtRetain	int	设定遗言讯息是否会被QP-500保留。
persistDir	char* / String	设定硬盘暂存机制所使用路径。 仅提供Windows版本MQTT SDK使用。
常数MQISDP_CLIENT_ID_LENGTH其值为23的长整数。		

## 2.2 函式:

### ICPDAS\_MQIsdpConnect:

`int ICPDAS_MQIsdpConnect ( PSPARMS *psp);`

函示功能说明 :

协助使用者透过此函式更简易地来连接上SmartQ QP-500。

参数说明 :

#### Inputs:

- `psp`:  
储存SmartQ QP-8x1/QP-8x2 or Windows Client与SmartQ QP-500连接所需的信息, PSPARMS结构定义请参考2.1节。

#### Returns:

- 回传整数值:
  - 0 : OK
  - 1002 : HOSTNAME NOT FOUND
  - 1007 : PERSISTENCE FAILED
  - 1010 : NO WILL TOPIC
  - 1013 : DATA TOO BIG
  - 1014 : ALREADY CONNECTED
- `psp`:  
如果ICPDAS\_MQIsdpConnect回传值为MQISDP\_OK(0), 则`psp->hConn`将回传为一个有效的内存地址, 反之`psp->hConn`将被设定为NULL。

---

### ICPDAS\_MQIsdpDisconnect:

`int ICPDAS_MQIsdpDisconnect (PSPARMS *psp );`

函示功能说明 :

协助使用者透过此函式更简易地中断与SmartQ QP-500联机。

参数说明 :

#### Inputs:

- **psp:**  
储存SmartQ QP-8x1/QP-8x2 or Windows Client与SmartQ QP-500连接所需的信息，PSPARMS结构定义请参考2.1节。

**Returns:**

- 回传整数值：
    - 0 : OK
    - 1007 : PERSISTENCE\_FAILED
    - 1008 : CONN\_HANDLE\_ERROR
  - **psp:**  
psp->hConn回传为NULL表示invalid connection 处理。
- 

**ICPDAS\_MQIsdpPublish:**

```
int ICPDAS_MQIsdpPublish ( PSPARMS *psp,  
                           char *pData,  
                           int dataLength );
```

函示功能说明：

协助使用者透过此函式更简易地经由SmartQ QP-500来发布讯息。

参数说明:

**Inputs:**

- **psp:**  
储存SmartQ QP-8x1/QP-8x2 or Windows Client与SmartQ QP-500连接所需的信息，PSPARMS结构定义请参考2.1节。使用者必须先将所欲发布的topic设定在psp->topic当中。
- **pData:**  
SmartQ QP-8x1/QP-8x2 or Windows Client欲发布与SmartQ QP-500的字符串数据。
- **dataLength:**  
欲发布字符串数据的长度。

**Returns:**

- 回传整数值：
  - 0 : OK
  - 1003 : Queue FULL
  - 1007 : PERSISTENCE FAILED
  - 1008 : CONN HANDLE ERROR
  - 1011 : INVALID STRUCTURE LENGTH

1013 : DATA TOO BIG  
1017 : CONNECTION BROKEN

- **psp:**

psp->lastSentMsg回传值:

1 : DELIVERED  
2 : RETRYING  
3 : IN PROGRESS  
4 : MSG HANDLE ERROR

---

## ICPDAS\_MQIsdpSubscribe:

[int](#) ICPDAS\_MQIsdpSubscribe ( **PSPARMS** \*psp );

函示功能说明 :

协助使用者透过此函式更简易地向SmartQ QP-500来订阅讯息。

参数说明 :

### Inputs:

- **psp:**

储存SmartQ QP-8x1/QP-8x2 or Windows Client与SmartQ QP-500连接所需的信息, PSPARMS结构定义请参考2.1节。使用者必须先将所欲订阅的topic设定在psp->topic当中。

### Returns:

- 回传整数值:

0 : OK  
1003 : Queue FULL  
1007 : PERSISTENCE FAILED  
1008 : CONNECTION HANDLE ERROR  
1011 : INVALID STRUCTURE LENGTH  
1013 : DAT TOO BIG  
1017 : CONNECTION BROKEN

- **psp:**

如果ICPDAS\_MQIsdpSubscribe回传值为OK(0), 则psp->lastSentMsg将回传一个有效值, 反之psp->lastSentMsg将被设定为(-1), 表示INVALID MSG HANDLE。

psp->lastSentMsg回传值:

1 : DELIVERED

- 2 : RETRYING
  - 3 : IN\_PROGRESS
  - 4 : MSG\_HANDLE\_ERROR
- 

## ICPDAS\_MQIsdpUnsubscribe:

`int ICPDAS_MQIsdpUnsubscribe ( PPSARMS *psp );`

函示功能说明 :

协助使用者透过此函式更简易地向SmartQ QP-500取消订阅讯息。

参数说明 :

### Inputs:

- **psp:**  
储存SmartQ QP-8x1/QP-8x2 or Windows Client与SmartQ QP-500连接所需的信息，PPSARMS结构定义请参考2.1节。使用者必须先将所欲取消订阅的topic设定在psp->topic当中。

### Returns:

- 回传整数值:
    - 0 : OK
    - 1003 : Queue FULL
    - 1007 : PERSISTENCE FAILED
    - 1008 : CONN HANDLE ERROR
    - 1011 : INVALID STRUCTUR LENGTH
    - 1013 : DATA TOO BIG
    - 1017 : CONNECTION BROKEN
  - **psp:**  
如果ICPDAS\_MQIsdpUnsubscribe回传值为OK(0)，则psp->lastSentMsg将回传一个有效值，反之psp->lastSentMsg将被设定为INVALID MSG HANDLE (-1)。  
psp->lastSentMsg回传值:
    - 1 : DELIVERED
    - 2 : RETRYING
    - 3 : IN PROGRESS
    - 4 : MSG\_HANDLE\_ERROR
-

## ICPDAS\_MQIsdpReceive:

```
int ICPDAS_MQIsdpReceive ( PSPARMS *psp,  
                           char *topic,  
                           char *data );
```

函示功能说明：

协助使用者透过此函式更简易地向SmartQ QP-500接收讯息。

参数说明：

### Inputs:

- **psp:**  
储存SmartQ QP-8x1/QP-8x2 or Windows Client 与SmartQ QP-500连接所需的信息，PSPARMS结构定义请参考2.1节。
- **topic:**  
用以储存SmartQ QP-8x1/QP-8x2 or Windows Client从SmartQ QP-500所接收到的讯息之主题。
- **data:**  
用以储存SmartQ QP-8x1/QP-8x2 or Windows Client从SmartQ QP-500所接收到的讯息内文信息。

### Returns:

- **回传整数值:**
  - 0 : OK
  - 1008 : CONNECTION HANDLE ERROR
  - 1005 : PUBS AVAILABLE
  - 1006 : NO PUBS AVAILABLE
  - 1018 : DATA TRUNCATED
- **topic:**  
用以储存SmartQ QP-8x1/QP-8x2 or Windows Client从SmartQ QP-500所接收到的讯息之主题。
- **data:**  
用以储存SmartQ QP-8x1/QP-8x2 or Windows Client从SmartQ QP-500所接收到的讯息内文信息。

---

## ICPDAS\_MQIsdpTerminate:

`void ICPDAS_MQIsdpTerminate(PSPARMS *psp);`

函示功能说明：

协助使用者透过此函式更简易地释放资源。

参数说明：

**Inputs:**

- **psp:**  
储存SmartQ QP-8x1/QP-8x2 or Windows Client与SmartQ QP-500连接所需的信息，PSPARMS结构定义请参考2.1节。

**Returns:**

- 无回传。

## 3. 范例程序说明

### 3.1 Publish.c (Publish\_WinCE.cpp)

#### 3.1.1 定义结构 PSPARMS。(PSPARMS 详细结构请见第二章节)

#### 3.1.2 设定变量:

##### (必要设定)

```
strcpy( psParams.clientId, "Client_Pub");  
// 设定SmartQ QP-8x1/QP-8x2 or Windows Client ID为"Client_Pub"。  
psParams.pBroker = (char*)malloc( strlen("192.168.100.63") + 1 );  
// SmartQ QP-500 IP位置字符串空间配置  
psParams.topic = (char*)malloc( strlen("MyTopic")+ 1 );  
// SmartQ QP-8x1/QP-8x2 or Windows Client 欲发布至SmartQ QP-500的讯息主题字符串空间配置  
strcpy( psParams.pBroker, "192.168.100.63");  
// 此程序目前暂定SmartQ QP-500 IP为192.168.100.63, 使用者可依实际状况调整IP。  
psParams.port = 1883;  
// 设定与SmartQ QP-500连接的通讯端口号码, 标准设置为1883。  
strcpy(psParams.topic, "MyTopic");  
// 设定讯息发布的主题为MyTopic。  
psParams.keepAliveTime = 32000;  
// 设定与SmartQ QP-500联机的存活时间为32000毫秒。
```

##### // persistDir仅提供Windows版本SmartQ SDK使用。

```
psParams.persistDir = (char*)malloc( strlen("C:\\temp\\wmqtt")+ 1 );  
// SmartQ Windows Client的硬盘暂存路径字符串空间配置。  
strcpy( psParams.persistDir, "C:\\temp\\wmqtt");  
// 设定与SmartQ Windows Client的硬盘暂存路径, 可依使用者需求修改路径。
```

##### (选择设定)

```
psParams.qos = 1;  
// 设定传输质量QoS为1, 如未设定默认值为0。  
psParams.retain = 0;  
// 设定发布信息于SmartQ QP-500部份不保留。  
psParams.lwtTopic = (char*)malloc( strlen("MyTopic")+ 1 );  
strcpy(psParams.lwtTopic, "MyTopic");  
// 为遗言主题配置内存空间并将其字符串设定为MyTopic, 如不使用遗言则可将其字符串设为NULL。  
psParams.lwtQos = 1;  
// 设定遗言的传输质量QoS为1, 如未设定默认值为0。  
psParams.lwtRetain = 0;
```



```
// 设定遗言讯息于SmartQ QP-500部份不保留。  
psParms.lwtData = (char*)malloc( strlen("Client_Pub terminated.")+ 1 );  
strcpy(psParms.lwtData, "Client_Pub terminated. ");  
// 为遗言数据配置内存空间并将其字符串设定为Client_Pub terminated.，如不使用遗言则可将其字符串设为NULL。。
```

### 3.1.3 联机：

```
ICPDAS_MQIsdpConnect( &psParms );  
// SmartQ QP-8x1/QP-8x2 or Windows Client 呼叫ICPDAS_MQIsdpConnect函式与SmartQ QP-500  
进行联机，传入参数为psParms，若回传值为0则联机成功，反之则联机失败并结束程序。
```

### 3.1.4 发布：

```
pBuffer = (char*)malloc(bufLen);  
// 为pBuffer配置内存空间  
fgets(pBuffer, bufLen, stdin);  
// 读取由为用户输入的讯息字符串。  
if ( msgLen <=0 ) {  
    ICPDAS_MQIsdpPublish( &psParms, "\0", 1 );  
    break; }  
// 当用户仅输入ENTER时程序结束，并通知SubScribe程序结束。  
ICPDAS_MQIsdpPublish( &psParms, pBuffer, msgLen );  
// SmartQ QP-8x1/QP-8x2 or Windows Client呼叫ICPDAS_MQIsdpPublish函式将Hello字符串发布与  
SmartQ QP-500，传入参数为psParms、pBuffer及msgLen。
```

### 3.1.5 中断联机：

```
ICPDAS_MQIsdpDisconnect( &psParms );  
// SmartQ QP-8x1/QP-8x2 or Windows Client 呼叫ICPDAS_MQIsdpDisconnect函式与SmartQ  
QP-500中断联机，传入参数为psParms。
```

### 3.1.6 释放内存：

```
ICPDAS_MQIsdpTerminate( &psParms );  
// SmartQ QP-8x1/QP-8x2 or Windows Client 呼叫ICPDAS_MQIsdpTerminate()，以结束所有工作及  
资源。  
free( pBuffer );  
// 释放变量所配置的内存空间。
```

## 3.2 SubScribe.c (SubScribe \_WinCE.cpp)

### 3.2.1 定义结构 PSPARMS。(PSPARMS 详细结构请见第二章节)

### 3.2.2 设定变量:

(必要设定)

```
strcpy( psParams.clientId, "Client_Sub" );
// 设定SmartQ QP-8x1/QP-8x2 or Windows Client ID为"Client_Sub"。

psParams.pBroker = (char*)malloc( strlen("192.168.100.63") + 1 );
// SmartQ QP-500 IP位置字符串空间配置

psParams.topic = (char*)malloc( strlen("MyTopic")+ 1 );
// SmartQ QP-8x1/QP-8x2 or Windows Client欲发布至SmartQ QP-500的讯息主题字符串空间配置

strcpy( psParams.pBroker, "192.168.100.63" );
// 此程序目前暂定SmartQ QP-500 IP为192.168.100.63, 使用者可依实际状况调整IP。

psParams.port = 1883;
// 设定与SmartQ QP-500连接的通讯端口号码, 标准设置为1883。

strcpy(psParams.topic, "MyTopic" );
// 设定讯息订阅的主题为MyTopic。

psParams.keepAliveTime = 32000;
// 设定与SmartQ QP-500联机的存活时间为32000毫秒。

psParams.lastSentMsg = MQISDP_INV_MSG_HANDLE;
// 预设处理状况为-1。
```

**// persistDir仅提供Windows版本SmartQ SDK使用。**

```
psParams.persistDir = (char*)malloc( strlen("C:\\temp\\wmqtt")+ 1 );
// SmartQ Windows Client的硬盘暂存路径字符串空间配置。

strcpy( psParams.persistDir, "C:\\temp\\wmqtt" );
// 设定与SmartQ Windows Client的硬盘暂存路径, 可依使用者需求修改路径。
```

(选择设定)

```
psParams.qos = 1;
// 设定传输质量QoS为1, 如未设定默认值为0。

psParams.timeout = 10000;
// 设定等待接收信息的时间为10000毫秒。

psParams.lwtTopic = (char*)malloc( strlen("MyTopic")+ 1 );
strcpy(psParams.lwtTopic, "MyTopic" );
// 为遗言主题配置内存空间并将其字符串设定为MyTopic, 如不使用遗言则可将其字符串设为NULL。

psParams.lwtQos = 1;
// 设定遗言的传输质量QoS为1, 如未设定默认值为0。

psParams.lwtRetain = 0;
```

```
// 设定遗言讯息于SmartQ QP-500部份不保留。  
psParms.lwtData = (char*)malloc( strlen("Client_Sub terminated.")+ 1 );  
strcpy(psParms.lwtData, "Client_Sub terminated. ");  
// 为遗言数据配置内存空间并将其字符串设定为Client_Sub terminated., 如不使用遗言则将其字符串  
// 设为NULL。。
```

### 3.2.3 联机:

```
ICPDAS_MQIsdpConnect( &psParms );  
// SmartQ QP-8x1/QP-8x2 or Windows Client呼叫ICPDAS_MQIsdpConnect函式与SmartQ QP-500进  
// 行联机, 传入参数为psParms, 若回传值为0则联机成功, 反之则联机失败并结束程序。
```

### 3.2.4 订阅:

```
ICPDAS_MQIsdpSubscribe( &psParms );  
// SmartQ QP-8x1/QP-8x2 or Windows Client呼叫ICPDAS_MQIsdpSubscribe函式以通知SmartQ  
// QP-500其订阅那些讯息主题, 传入参数为psParms。
```

### 3.2.5 接收:

```
rcvTopic = (char*)malloc( BUFFER_SIZE );  
rcvData = (char*)malloc( BUFFER_SIZE );  
// 为rcvTopic及rcvData配置内存空间。  
ICPDAS_MQIsdpReceive( &psParms, rcvTopic, rcvData );  
// SmartQ QP-8x1/QP-8x2 or Windows Client呼叫ICPDAS_MQIsdpReceive函式进行订阅主题的讯息  
// 接收, 传入参数为psParms、rcvTopic及rcvData, 并经由rcvTopic及rcvData取得所接收到的主题及  
// 信息。  
// 若psParms.timeout = -1; 则此函式会在有接收到主题信息后才会返回 (仅于Multi Task下支援)  
// 若psParms.timeout 其它数据(单位为ms); 则此函式会在有接收到主题信息或Timeout时间到达而返  
// 回
```

### 3.2.6 取消订阅:

```
ICPDAS_MQIsdpUnsubscribe( &psParms );  
// SmartQ QP-8x1/QP-8x2 or Windows Client呼叫ICPDAS_MQIsdpUnsubscribe函式以通知SmartQ  
// QP-500其取消订阅那些讯息主题, 传入参数为psParms。
```

### 3.2.7 中断联机:

```
ICPDAS_MQIsdpDisconnect( &psParms );  
// SmartQ QP-8x1/QP-8x2 or Windows Client呼叫ICPDAS_MQIsdpDisconnect函式与SmartQ QP-500  
// 中断联机, 传入参数为psParms。
```

### 3.2.8 释放内存:

```
ICPDAS_MQIsdpTerminate ( &psParms );  
// SmartQ QP-8x1/QP-8x2 or Windows Client 呼叫ICPDAS_MQIsdpTerminate(), 以结束所有工作及  
// 资源。  
free( rcvData );  
free( rcvTopic );
```

// 释放变量所配置的内存空间。