

## ETA6210 (WINCE) Wifi 扩展模块应用手册

感谢您购买英创信息技术有限公司的产品：**ETA6210(WINCE)Wifi扩展模块**。

您可以访问英创公司网站或直接与英创公司联系以获得ETA6210的其他相关资料。

英创信息技术有限公司联系方式如下：

地址：成都市高新区高朋大道5号博士创业园B座407# 邮编：610041

联系电话：028-86180660 传真：028-85141028

网址：<http://www.emtronix.com> 电子邮件：[support@emtronix.com](mailto:support@emtronix.com)

# ETA6210 简介

ETA6210 是一款标准 miniPCle 尺寸的 WiFi 通讯模块，模块核心芯片为 NXP 的 iMX6UL 和 AP6181 WiFi 芯片。英创公司 WINCE 工控主板是通过 miniPCle 上的 USB 接口来操作 ETA6210 模块的。在实际应用中，ETA6210 是插在客户底板的 miniPCle 插座上使用。开发评估阶段，可插在英创的 ESMARC EVB 评估底板的 miniPCle 插座上使用，也可利用 ETA303 模块转换成标准 USB 接口使用。

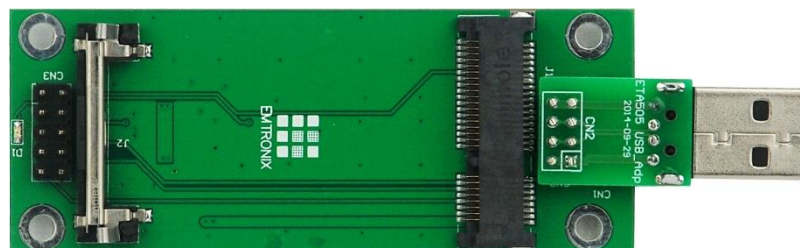
ETA6210 本身预装有完整的操作系统，在上电后自动进行 WIFI 连接。并同时打开 3 路虚拟串口 COM10-12 与 WINCE 工控主板进行通信:一路(COM10)用 AT 指令进行参数设置，模块控制等。一路 (COM11) 用于网络透传。一路(COM12)备用。

ETA6210 网络模式与常规 WIFI 模块略有区别。ETA6210 直接连入 WIFI 网络，同时作为路由器接受嵌入式主板的连接。ETA6210 将 WIFI 网络中的数据直接透传给嵌入式主板，所以对于嵌入式主板而言，与直接连入 WIFI 网络没有区别，只是无法直接调用 API 函数获得 WIFI 网络中的 IP, MASK, 网关，而是需要通过 AT 指令访问 ETA6210 获得。

连接到 WIFI 网络后，可以使用 SOCKET 编程访问网络。

# 安装

ETA6210 可以通过 miniPCle 直接连接到英创 ESMARC 底板上，也可以通过 ETA303 模块转接到板子 USB 上。图 1 和图 2 显示它们的连接关系。



图、ETA303





连接成功后会有成功提示。

英创提供 APConn 程序源码，客户可以根据自身需要进行修改。

## 接口介绍

英创提供的例程 ETA6210Set 是对 ETA6210 模块进行配置的示例例程。例程将 ETA6210 模块接口封装到了 class ETA6210 类中，ETA6210 主要 PUBLIC 接口如下：

```
class ETA6210
{
public:
    ETA6210(void);
    ~ETA6210(void);
    BOOL CreateEntry();
    BOOL GetState(LinkStates *plinkStates);
    BOOL GetAPList(APInfo *papInfo, int maxnum, int *pnnum);
    DWORD ConnectAP(char *ssid, char *password, BOOL bDhcp, IPInfo
    *IpInfo);
    BOOL DisconnectAP();
    BOOL GetIPInfo(IPInfo *pIpInfo);
    BOOL SetIPInfo(IPInfo *pIpInfo);
};
```

客户将封装好的库添加进自己工程中，即可很方便的操作 ETA6210。

参考例程，使用时在工程中添加 Serial.h, Serial.cpp, ETA6210.h, ETA6210.cpp 共 4 个文件。并定义 ETA6210 类

```
#include "ETA6210.h"
```

```
ETA6210 eta6210;
```

## ETA6210 模块初始化

使用 ETA6210 首先需要创建拨号连接。拨号连接只需要创建一次，重复创建不会产生变化，不会返回失败。

### 函数原型

```
BOOL CreateEntry();
```

### 返回值

调用成功返回 TRUE，失败返回 FALSE。

### 调用实例

```
bRet = eta6210.CreateEntry();
```

## 查询 wifi 列表

查询 ETA6210 能够搜索到的附近 AP，并将它们存入结构体数组中返回。

### 函数原型

```
BOOL GetAPList(APInfo *papInfo, int maxnum, int *pnum);
```

### 参数

**papInfo** 指向 APInfo 结构体数组指针，当调用成功，AP 信息将依次存入结构体数组中。

**maxnum** 传入的结构体数组大小。

**pnum** 返回的 AP 数量，不会超过 maxnum。

### 返回值

调用成功返回 TRUE，失败返回 FALSE。

### 参数结构体定义

```
struct APInfo
```

```

{
    char    ssid[50];        //ssid
    char    mac[18];        //mac ca:d7:19:d8:a6:44
    int     channel;
    int     rssi;           //rssi 信号强度
    int     ecn;            //加密方式: OPEN, WEP, WPA_PSK, WPA2_PSK,
WPA_WPA2_PSK
    int     freqoffset;     //频偏
    int     freqcalibration; //频率校准
};

```

### 调用实例

```

APIInfo apInfo[15];
int n;
bRet = eta6210.GetAPList(apInfo, 15, &n);

```

## 连接 WIFI

设置要连接的 WIFI 热点的 SSID，及密码，并拨号连接。

### 函数原型

```

DWORD ConnectAP(char *ssid, char *password, BOOL bDhcp, IPInfo
*IpInfo);

```

### 参数

**ssid** AP 名称，如果为空，则使用上次连接的 SSID。

**password** AP 密码，如果为空，则使用上次使用的密码。

**bDhcp** AP 是否使用 DHCP 进行 IP 分配，对于无 DHCP 的 AP，此参数必须为 TRUE，并设置 IpInfo 参数里的 IP 信息。

**IpInfo** 当连接无 DHCP 的 AP 时，必须设置 IP 及 MASK 参数。当连接 DHCP 的 AP 时，此参数无效，可以设置为 NULL。

### 返回值

连接为 DWORD，定义如下：

- 0 成功
- 1 设置 SSID 失败

- 2 设置密码失败
- 3 连接 AP 失败
- 4 AP 不支持 DHCP，且未设置 IP
- 5 设置 IP 失败
- 6 拨号失败

#### 调用实例

```
bRet = eta6210.ConnectAP("emtronix.20", "0987654321", FALSE,  
NULL);
```

或

```
bRet = eta6210.ConnectAP(NULL, NULL, FALSE, NULL); //使用上  
次的 SSID 及密码
```

## 断开连接

断开 WIFI 连接。

#### 函数原型

```
BOOL DisconnectAP();
```

#### 返回值

成功断开返回 TRUE，失败返回 FALSE。

#### 调用实例

```
bRet = eta6210.DisconnectAP();
```

## 查询模块状态

查询模块当前状态。包括模块是否处于连接状态，连接的 AP 的 SSID。

#### 函数原型

```
BOOL GetState(LinkStates *plinkStates);
```

#### 参数

plinkStates 指向 LinkStates 结构体，记录模块网络是否连接成功，及



连接的 SSID 名，及信号强度。

### 返回值

调用成功返回 TRUE，失败返回 FALSE。

### 参数结构体定义

```
struct LinkStates
{
    BOOL    bLink;           //连接状态
    char    ssid[50];       //ssid
    int     rssi;           //信号强度
}
```

### 调用实例

```
LinkStates linkStates;
bRet = eta6210.GetState(&linkStates);
```

## 查询网络参数

因为英创嵌入式主板只连接到 ETA6210 模块的内部网络中，ETA6210 再连接到 WIFI 网络并将数据透传给英创嵌入式主板。嵌入式主板通过 API 只能获得内部网络里的 IP，必须通过以下接口函数获得在 WIFI 网络里的 IP，MASK，网关参数。

### 函数原型

```
BOOL GetIP(IPInfo *pIpInfo);
```

### 参数

**pIpInfo** 指向 IpInfo 结构体，储存有连接到 WIFI 中时的网络 IP，mask 及网关。

### 返回值

调用成功返回 TRUE，失败返回 FALSE。

### 参数结构体定义

```
struct IPInfo
{
    char ip[16];
    char gw[16];
    char mask[16];
};
```

### 调用实例

```
IPInfo ipInfo;  
bRet = eta6210.GetIP(&ipInfo);
```

## 设置网络参数

设置模块连入 WIFI 网络里的网络参数，如，在做 TCP 服务器时需要用到。

### 函数原型

```
BOOL SetIP(IPInfo *pIpInfo);
```

### 参数

**pIpInfo** 指向 IpInfo 结构体，将要设置的网络 IP, mask 及网关填入结构体中，如果为空字符串“”，则该项不设置，任使用原参数。

### 返回值

调用成功返回 TRUE，失败返回 FALSE。

### 参数结构体定义

```
struct IPInfo  
{  
    char ip[16];  
    char gw[16];  
    char mask[16];  
};
```

### 调用实例

```
IPInfo ipInfo;  
bRet = eta6210.GetIP(&ipInfo);  
sprintf(ipInfo.ip, "192.168.201.248");  
sprintf(ipInfo.gw, "192.168.201.19");  
sprintf(ipInfo.mask, "255.255.255.0");  
bRet = eta6210.SetIP(&ipInfo);
```

## SOCKET 编程示例

拨号成功后，可以使用 SOCKET 进行网络通信。

以英创开发光盘里的例程为例。

## TCP 客户端 C#例程

参考英创开发光盘中 C#例程 step4\_tcptest 例程，不用修改例程即可直接进行调试。

### 连接服务端

通过 WIFI 网络，连接到服务器，并建立接收线程

```
try
{
    rip = IPAddress.Parse(textBox_rip.Text);           //IP
    rport = Convert.ToInt32(textBox_rport.Text);      //端口
    IPEndPoint rEP = new IPEndPoint(rip, rport);      //用
    远程IP, 端口初始一个IPEndPoint类

    cTcp = new TcpClient();
    cTcp.Connect(rEP);
    revThread = new Thread(new ThreadStart(waitforMessage));
    threadStop = false;
    revThread.Start(); //启动waitforMessage线程
}
catch(Exception ex)
{
    MessageBox.Show("连接失败: ");
    statusBar1.Text = (ex.Message);
    return;
}
```

### 发送消息

```
byte[] OutBuffer;
int length;
OutBuffer = Encoding.Default.GetBytes(textBox_send.Text);
```

```
length = OutBuffer.Length;  
cTcp.Client.Send(OutBuffer, length, 0);
```

## 消息接收线程函数

```
void waitforMessage() //客户机状态下接收数据线程  
{  
    int i;  
    byte[] InBuffer = new byte[1024];  
    while (!threadStop)  
    {  
        i = cTcp.Client.Receive(InBuffer);  
        if (i == 0)  
        {  
            Invoke(new EventHandler(serverDisconnect));  
            //注意使用Invoke  
            return;  
        }  
        revstr = Encoding.Default.GetString(InBuffer, 0,  
InBuffer.Length);  
        Invoke(new EventHandler(textBox_rev_Show));  
    }  
}
```



图、光盘 C#例程 step4\_tcptest 测试图

## TCP 服务端 C 例程

参考英创开发光盘中 C 例程 test\_tcp\_server 例程。并对例程稍作修改

### 获得 WIFI 口 IP

获得 WIFI 网口的 IP，并显示到 MFC 程序界面上。

```
m_CEdit_Name2.SetWindowText(L"ETA6210");
bRet = GetNetWorkAdapterInfo( L"ETA6210", &AdptInfo );
m_IP2 = AdptInfo.IPAddr;
wsprintf(wstr, L"%d.%d.%d.%d", m_IP2>>24, (m_IP2>>16)&0xff,
(m_IP2>>8)&0xff, m_IP2&0xff);
m_CEdit_IP2.SetWindowText(wstr);
m_CEdit_Port2.SetWindowText(L"5000");
```

### 打开服务端

初始化网络，创建 SOCKET 并绑定到 WIFI 网口，建立接收线程。

```
//1. 初始化socket资源
if (WSAStartup(MAKEWORD(1,1), &wsa) != 0)
{
    return -1; //代表失败
}

//2. 创建套接字
if ((m_socketServer = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP)) < 0)
{
    return -2;
}

//3. 绑定端口
SOCKADDR_IN serverAddr;
ZeroMemory((char *)&serverAddr, sizeof(serverAddr));
serverAddr.sin_family = AF_INET;
serverAddr.sin_port = htons(m_dwPort);
```

```
serverAddr.sin_addr.s_addr = htonl(m_dwIP);
if (bind(m_socketServer, (struct sockaddr *)&serverAddr,
sizeof(serverAddr))<0)
{
    return -3 ;
}

//4. 设置为侦听套接字, 可同时响应个客户的连接请求
if (listen(m_socketServer, 8)!=0)
{
    return -4;
}

//4. 设置监听套接字通讯模式为异步模式
//DWORD u1= 1;
//ioctlsocket(m_socketServer, FIONBIO, &u1);

//5. 创建侦听通讯线程, 在线程里, 等待客户端连接请求
m_bListenThreadStop = FALSE;
m_hListenThread = CreateThread(NULL , 0, SocketListenThread, this,
0, NULL);
if (m_hListenThread == NULL)
{
    closesocket(m_socketServer);
    return -5;
}
```

## 发送消息

```
iLen = m_CEdit_Send1.GetWindowText(wstr, 1024);
iLen = WideCharToMultiByte(CP_ACP, 0, wstr, iLen, str, 1024, NULL,
NULL);
m_TCPServer1.Send(str, iLen);
```

## 消息接收线程函数

```
DWORD CTCPServer::SocketListenThread(PVOID lparam)
{
    CTCPServer *pTCPServer;
    //得到CTCPServer_CE实例指针
    pTCPServer = (CTCPServer*)lparam;
    while (!pTCPServer->m_bListenThreadStop)
    {
```

```
SOCKADDR_IN clientAddr;
int addrLen = sizeof(clientAddr);
//等待客户端连接
pTCPServer->m_socketClient[0] =
accept(pTCPServer->m_socketServer, (struct sockaddr *)&clientAddr,
&addrLen);
if (pTCPServer->m_socketClient[0] > 0) //判定是否连接上了
{
    pTCPServer->m_revLen = recv(pTCPServer->m_socketClient[0],
pTCPServer->m_revbuf, sizeof(pTCPServer->m_revbuf), 0);
    while (pTCPServer->m_revLen > 0)
    {
        pTCPServer->OnRev(lparam);
        pTCPServer->m_revLen =
recv(pTCPServer->m_socketClient[0], pTCPServer->m_revbuf,
sizeof(pTCPServer->m_revbuf), 0);
    }
}
return 0;
}
```

注：该例程设定的网口 2 由有线网口改为了 wifi，所以网络名修改为了“ETA6210”，同时绑定的 IP 需要设置为 ETA6210 内部网络的 IP，默认“192.168.1.2”。但是本例测试时，连入到 WIFI 网络中的 IP 通过接口函数 GetIP 查到是“192.168.201.112”，所以测试 PC 连接的 TCP 服务端 IP 为“192.168.201.112”而不是“192.168.1.2”



图、光盘 C 例程 test\_tcp\_server 测试图

## 网络测试

WIFI 网络内的 PC 可以通过 IP PING 通使用 ETA6210 的嵌入式主板，也可以通过 TELNET 连接到嵌入式主板中。

测试 ETA6210 可以满足正常的 WIFI 应用。

英创提供了例程及源代码，有需要的客户可以联系英创工程师获得。