



嵌入式网络模块在无线上网中的应用

目 录

前 言	3
第一部分 系统构成	4
应用单元与系统的关系	4
IP 地址配置	5
第二部分 TCP/IP 通讯接口函数及其应用	7
接口函数说明	7
应用案例及其实现	13
第三部分 通过 PPP 访问互联网	17
PPP 协议接口函数说明	17
应用案例及其实现	19
第四部分 标准 MODEM 拨号的实现	24
第五部分 通过 GPRS 模块进入互联网	25
第六部分 通过无线模块收发短消息	26
第七部分 应用程序设计说明	27

前 言

无线移动通讯网络（GSM）的发展，特别是 GPRS 的出现，为工业自动化设备的上网提供了新的手段。通过 GPRS 服务，设备可采用互联网（internet）的标准方式，如收发 email、FTP 文件传输等等，与在互联网上的数据中心交换数据。众所周知，互联网的通讯是建立在 TCP/IP 协议之上的，因此设备还可以与自己的数据中心直接建立 TCP 连接，进行用户定义的数据通讯。

当前，以局域网（LAN）为标志的网络通讯技术已在工业自动化领域广泛应用，通常在一个逻辑网络中包括多个设备，这些设备都可能需要与互联网上的数据中心交换数据，这就需要一种既能与 LAN 上的设备进行通讯，又能够根据需要访问互联网上的不同资源的类似网关的设备。

采用英创公司的嵌入式网络模块与通用的无线通讯模块相结合，可方便的实现智能设备访问互联网的目的。英创公司的嵌入式网络模块是以 Intel 的高性能 32-bit 嵌入式微处理器 386EX 为核心的 CPU 板级模块系列产品，包括 NetBox、ETR232E 等若干产品型号。硬件方面这些模块均配备了完整的网络通讯接口，以及面向智能化设备的键盘显示接口和扩展总线。软件方面则配备了丰富的支持软件，包括 TCP/IP 协议库、串口驱动、汉字显示、看门狗及低层配置程序等等。

本文主要是从应用软件开发的角度介绍如何通过嵌入式网络模块提供的软硬件资源，同时与以太网和 GPRS 模块相连接，实现相应的数据处理及传送。从而为需要无线上网的广大客户提供一个较为完整的应用方案。本文共包括 6 个部分，第一部分介绍系统的构成，及各个子系统之间的关系；第二部分主要介绍嵌入式网络模块与以太网上设备的通讯；第三部分则重点介绍基于 PPP 的互联网登录及在互联网上的常规通讯；第四部分提供了通过标准有线 Modem 上网的方法；第五部分介绍通过 GPRS 无线模块上网的方法；第六部分主要对应用程序设计中的有关事项进行概括说明。

第一部分 系统构成

应用单元与系统的关系

英创嵌入式网络模块实现无线上网是针对英创公司的嵌入式网络模块在无线通讯应用的需求提出的。典型的应用系统由一块嵌入式网络模块（NetBox、ETR232E、NB200、NB104 之一）和一块具有 GPRS 无线数据通讯能力的 GSM 模块组成。一般的 GPRS 模块都有一个标准是 RS232 串口，供终端设备与之相连，英创嵌入式网络模块正是通过串口与 GPRS 模块实现通讯的，客户的应用程序将在英创嵌入式网络模块中运行。

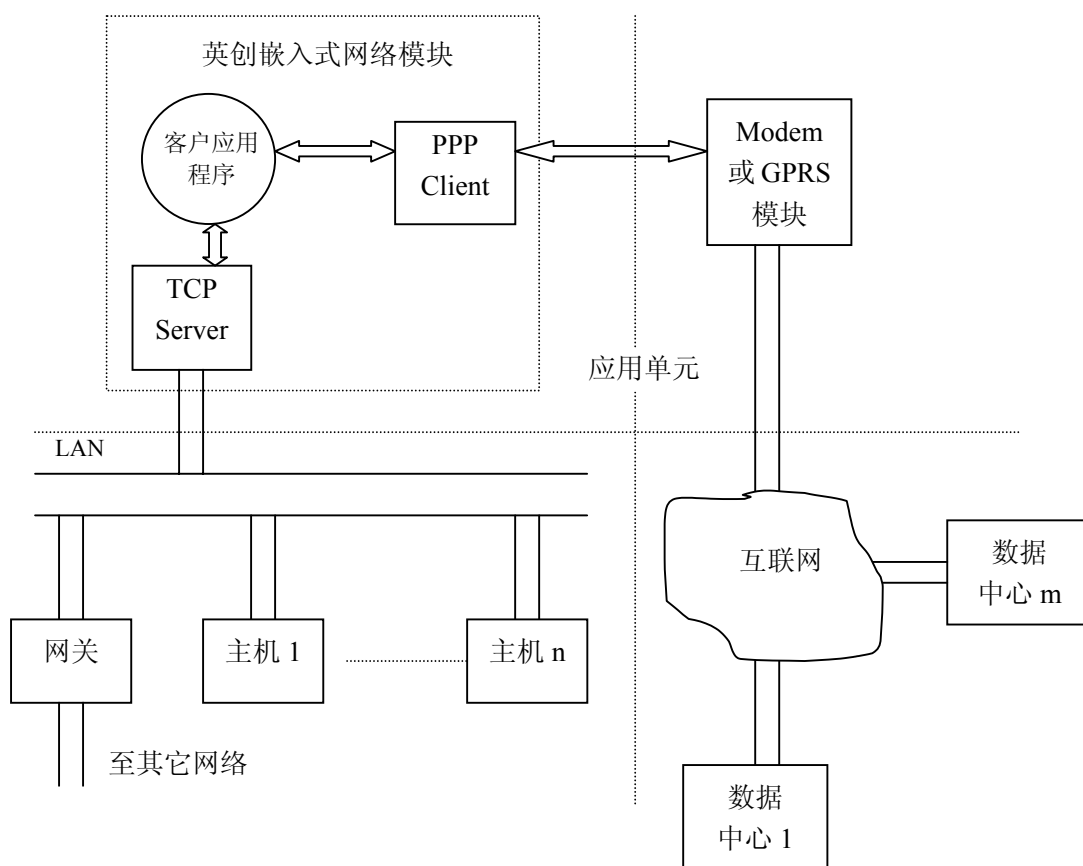


图 1 基于英创嵌入式网络模块的无线通讯应用方案框图

在本文的具体实现中，嵌入式网络模块采用的是 ETR232E, 在英创的嵌入式网络模块产品系列中 ETR232E 具有最低的成本，体积也是最小，而 NetBox 等其他模块则具有更强的扩展性。对简单的数据采集，网络转换方面的应用，采用 ETR232E 较为合适，而对智能终端方

面的应用，采用 NetBox 等具有更强扩展能力的模块能保证整机性价比的最优化。在本案例中的 GPRS 模块是 SIEMENS 的 MC35。MC35 是一款完整的具有 GPRS 功能的移动通讯模块，提供有完备的 AT 指令集对其进行操作控制。除了 GPRS 功能外，它还能作为标准手机，实现语音通话、短信（SMS）等功能，这些功能已超出了本文涉及的范围，有兴趣的用户可参考相关的技术资料。在本文的以下部分，我们将不加区别的使用嵌入式网络模块、ETR232E 和 NetBox；GPRS 模块和 MC35。

图 1 是以应用软件的观点描述的系统关系图，英创嵌入式网络模块一方面通过板上的串口 COM1 与 GPRS 模块相连，另一方面通过标准以太网接口与本地应用系统相连，嵌入式网络模块的主要功能是在应用层上相互转发 GPRS 无线网络与本地 LAN 的数据包，并可根据应用需求，对数据进行必要的处理，如加密、解密等。英创嵌入式网络模块都带有一个 TTL/COMS 电平的 RS232 接口，只需简单的电阻分压就可与 MC35 的 3V 电平的串口相连。

从网络结构的角度看，GPRS 网络的结构可表述为 TCP+IP+PPP+无线链路，而局域网 LAN 的结构可表述为 TCP+IP+ETHERNET+以太链路，由此反映出嵌入式网络模块作为网络路由的本质。

IP 地址配置

从应用的角度看，对本地局域网来说由 ETR232E 和 MC35 构成的单元是一个标准的 TCP 服务器，根据应用的需求，还可以把它升级为 FTP 服务器、Web 服务器等等。对外部互联网来说，ETR232E_MC35 单元则是一个通过 GPRS 无线方式接入的客户端，它采用点对点的方式，通过中国电信的 APN 访问互联网。无论是本地局域网的数据传送还是对互联网的访问，都是建立在 TCP/IP 协议上的，因此对 ETR232E 和 MC35 构成的单元需要有两个 IP 地址，一般来讲本地局域网的 IP 地址可通过配置文件的形式静态设置，如果需要服务器支持跨网段的支持，还需设置网关地址。而访问互联网的所用的 IP 则是通过 PPP 的协商，由 GPRS 的节点服务器动态分配给嵌入式网络模块的。注意本地网络的 IP 与动态分配获得的 IP 不能等同对待，即它们是两个完全不同的网络世界。本应用方案把与 GPRS 的节点服务器建立的 PPP 通道作为缺省的网关，即对 IP 对象的访问，都将通过 PPP 通道，而不会作为本地局域网的对象，这意味着所访问的 IP 都是互联网上的主机设备。

众所周知，互联网上的主机一般都有域名，应用程序如果需要通过域名访问主机，还需要设置一个域名服务器 IP，通过域名解析函数来转换成相应的 IP 地址。本应用方案已提供

了对域名解析的支持。

在本应用方案中，以服务器方式或被动方式打开的 TCP 连接，将响应本地局域网的数据请求，TCP 服务器支持同时打开多个连接，而以客户端或主动方式打开的 TCP 或 UDP 连接，都将通过 PPP 链接指向互联网上的主机设备。本方案还提供了在互连网上一些典型应用，如 FTP 文件传送、电子邮件的收发等等。有关网络操作的接口函数将在后续章节详细介绍。

第二部分 TCP/IP 通讯接口函数及其应用

英创嵌入式网络模块带有标准的 10Mbps 以太网接口，通过调用配套的 TCP/IP 协议库，可方便地与本地局域网以及通过网关连接的主机建立 TCP 或 UDP 的连接，进而实现数据通讯的功能。具体的实现的方法是用户在其应用程序中直接调用提供的通讯接口函数。本方案提供的所有通讯函数库均在 BC3.1 集成编程环境下生成的，因此用户的应用程序也应在 BC3.1 集成编程环境中编写。

大部分有关以太网的通讯接口函数与英创提供的常规配套的 TCP/IP 协议库中相应函数的调用形式及功能是完全一样的，主要区别在于许多函数即可用于以太网的通讯，又可用于基于 PPP 的无线网络的通讯，这是因为从应用层看，都是在 TCP 层上进行数据读写。以下对有关接口函数做详细的介绍。

接口函数说明

```
int InitEthernetNet( char* IPString=NULL, char* MSKString=NULL,
                    int COM_Port=0);
```

功能描述：初始化所有网络接口，包括以太网接口及 PPP 接口。

输入参数	描述
char* IPString	本地局域网 IP 地址串，形式为：“n1.n2.n3.n4”；IP 串为空，则系统按缺省 IP 地址进行初始化，缺省 IP 为：“192.168.201.22”。
char* MSKString	用户根据需要设置子网掩码，形式为：“m1.m2.m3.m4”。当 MSKString 串为空时，系统将按 IP 地址的类型设置其子网掩码。用户也可根据实际应用需要，设置相应的子网掩码，比如为 B 类 IP “162.168.201.22”，设置 C 类子网掩码“255.255.255.0”。用户可通过子网掩码的有效设置，来提高网络 IP 地址的利用率。
int COM_Port	与无线模块连接的串口号，COM_Port 的缺省配置为 0，表示

	使用串口 1 (COM1), COM_Port=1 时, 则使用串口 2 (COM2)
--	---

返回值: 若正确返回值为 0, 若初始化失败返回值<0。

备注:

- (1) 此函数调用时可以不填 IPString 而使用缺省值; MSKString 也可以不填 MSKString, 使用缺省值; 但不能不填 IPString 而直接只填 MSKString。
- (2) 本函数将同时对基于串口 (COM1) 的 PPP 网络进行初始化, 由于 PPP 网络的 IP 地址是由 PPP 服务器动态分配的, 所以不需要设定 IP。
- (3) 为了保持与常规配套 TCP/IP 库的一致性, 初始化函数名称不变。
- (4) 由于一般的 GPRS 模块的串口电平都是 3V 的, 因此更易于与 TTL/COMS 电平的串口连接。对 ETR232、ETR232E 建议使用 COM1 口, 而对 NetBox、NB200、NB104 则使用 COM2 口。

int TermEthernetNet();

功能描述: 关闭所有网络接口, 包括以太网接口和 PPP 接口。

返回值: 若正确返回值>0, 若失败返回值<0。

参数: 无

int ConnOpen (char * to, char * protoc, int lp, int rp, int flags);

功能描述:

打开 TCP 或 UDP 的连接。对局域网而言, 应采用被动打开 (服务器) 方式; 对 PPP, 则应采用主动打开 (客户端) 方式。

输入参数	描述
char* to	指定远方主机名称, 若以服务器方式打开, 则填 “*”; 以客户端方式打开, 则填远方主机的 IP 地址, 如 “192.168.201.97”。注意主动打开所引用的 IP 地址指的是互联网上的主机, 而不是本地局域网的主机。
char* protoc	定义传输层协议, 可取: ” TCP/IP” 或 ” UDP/IP”
Int lp	本地端口号。在服务器模式应用中, lp 为知名端口号。在一般应用程序中, 建议端口号大于 1000。
Int rp	若应用程序工作在服务器模式下, rp 应置为 0, 若应用程序工作在客户端模式下, rp 为连接对方端口号, 即知名端口号。

Int flags	一般置为 0。但在单任务的运行环境中的 TCP 连接，宜采用无阻塞方式打开连接，这时 flags 应置为 NONBLOCKOPEN。对无阻塞方式打开的连接，返回的连接号并不表示连接已建立，因此应用程序需通过调用 ConnIsEstablished(int conno) 来确定连接是否建立。
-----------	---

返回值	描 述
>=0	表示正确返回，其值为连接号，应用程序操作 TCP/IP 通讯接口时，都需使用该连接号
EHOSTUNRECH (-11)	远方主机不能访问
ENOBUFS (-15)	网络无数据缓冲区
ETIMEOUT (-12)	超时出错
ECONNABORTED (-14)	协议错误

备注：

- (1) 本函数对以太网和 PPP 均有效。
- (2) 对以太网只能以服务器方式打开连接。
- (3) 对 PPP 只能以客户端方式打开连接。

int ConnClose(int conno);

功能描述：关闭连接。

输入参数：连接号。

返回值	描 述
0	正常关闭
EBADF (-16)	连接号无效，关闭操作未执行
ECONNABORTED (-14)	协议错误，本次连接的数据传送有可能不可靠，连接已关闭

备注：

- (1) 未建立的连接进行关闭操作，可能会有 2 分钟左右的延时。
- (2) 如果当前打开的连接为 TCP 连接，关闭以后 1 分钟左右方能再使用。

int ConnRead(int conno, char* buff, int len);

功能描述：从打开的连接中读取数据。

输入参数	描 述
int conno	已打开的连接号
char* buff	Char 型指针变量，为存放读出数据的缓冲区 buff 地址
int len	表明存放数据缓冲区的字节长度

返回值	描 述
0	连接已被关闭；
>0	正常返回，返回值为实际读取的字节数；对 TCP 连接，最大缓冲区长度应不大于 1460 字节；对 UDP 来说，最大缓冲区长度应不大于 1472 字节。
EBADF (-16)	无效连接号，读取操作未执行。
EWOULDBLOCK (-18)	操作被阻塞，对无阻塞方式打开的连接，可重复调用该函数。
ETIMEOUT (-12)	超时，由应用程序决定后续处理。
EMSGSIZE (-19)	数据太长。

```
int ConnWrite( int conno, char* buff, int len, int PushFlg=0 );
```

功能描述：发送缓冲区中的数据至 TCP/IP 网络接口或至提供 UDP 服务的指定端口。

输入参数	描 述
Int conno	已打开的连接号
char* buff	char 型指针变量，为存放发送数据的缓冲区 buff 地址
Int len	表明存放数据缓冲区的字节长度
Int PushFlg	0：为普通发送 1：为急迫发送

返回值	描 述
0	连接已被关闭；
>0	正常返回，返回值为实际发送的字节数；对 TCP 连接，最大缓冲区长度应不大于 1460 字节；对 UDP 来说，最大缓冲区长度应不大于 1472 字节。
EBADF (-16)	无效连接号，发送操作未执行。

EWouldBlock (-18)	操作被阻塞,对无阻塞方式打开的连接,可重复调用该函数。
ETimeout (-12)	超时, 由应用程序决定后续处理。
EMSGSize (-19)	数据太长。

备注:

在普通发送时,如果需要连续发送的数据包小于最大缓冲区长度,而且在连续发送之间没有作相应的延时, TCP/IP 则有可能将连续的包作合包处理后发送,以提高网络的传输效率,所以在接收方收到的是已经合过的包。在众多工业应用中,可以通过设置急迫标志,并在数据包与数据包之间作短暂的延时(如 1ms)就可以使接收方收到与发送方一致的包。

`int ConnIsEstablished(int conno);`

功能描述: 检查连接是否建立。

输入参数: 连接号。

返回值	描 述
1	连接已建立
0	连接还未建立

`int ConnCanSend(int conno, int len);`

功能描述: 检查是否能发送指定长度的数据。

输入参数	描 述
Int conno	已打开的连接号
Int len	需发送数据的长度

返回值:

- 1: 接已能发送指定长度的数据;
- 0: 连接不能发送指定长度的数据。

`int ConnHasData(int conno);`

功能描述: 检查是否有数据可读取。

输入参数: 连接号。

返回值	描 述
1	连接已存在可读取的数据
0	连接无可读取的数据

`int ConnIsFinished(int conno);`

功能描述：当应用程序工作在服务器模式时，可使用该函数检查是否对方已关闭连接。

输入参数：连接号

返回值	描 述
1	连接已关闭
0	连接未关闭

`int JoinMulticastGroup (char* MulticastIP);`

功能描述：

本系统加入组播(且组播 IP 地址被 MulticastIP 唯一标示)，使本系统能收到向本组播 IP 地址发送的信息。(注:不能同时加入两个及以上的组播，要加入新的组播必须先退出上一组播)

输入参数： MulticastIP 组播的 IP 地址.

返回值	描 述
<0	该函数调用失败
>=0	该函数调用成功

备注：

- (1) 组播 IP 地址的设置需在打开网络连接之前进行。
- (2) 本函数只在需接收组播数据的场合下调用，组播的发送与发送标准的 UDP 数据包的方式一样。
- (3) 应用程序不能对一组播 IP 同时进行收发。

`int LeaveMulticastGroup ();`

功能描述：退出已加入的组播

返回值：若成功返回 0；若失败返回 -1

备注：退出组播需在相应网络连接关闭以后进行。

`int SetGateWayIP(char* GateWayIP) ;`

功能描述：

在不同局域网之间通讯时需要用到此函数设置网关地址。使本机可以通过网关路由

到另外的局域网的远方主机。

输入参数: char* GateWayIP 网关的 IP 地址.

返回值	描 述
<0	该函数调用失败
>=0	该函数调用成功

备注: 设置网关需在网络初始化之前进行。

```
int GetIP( int conno, unsigned char* IPCode );
```

功能描述: 获取连接对方的 IP 地址

输入参数	描 述
int conno	已打开的连接号
unsigned char* IPCode	当前连接的对方 IP 地址

返回值	描 述
<0	该函数调用失败
>=0	该函数调用成功

备注: 此函数只对已建立连接有效。对于以太网和 PPP 都可使用。

```
void NetPackagePro()
```

功能描述:

处理低层的网络数据包。应用程序如果在很长的一段时间内没有涉及到网络这一层, 应调用该函数, 使得网络数据包能够得到及时处理。

应用案例及其实现

在我们提供应用方案中, 以太网的通讯方式是以服务器方式或被动方式打开的 TCP 连接, 并支持同时打开多个连接, 响应本地局域网的数据请求。我们定义了一个 TCPServer 的类, 来实现以上功能。下面将说明 class TCPServer 的定义。

class TCPServer 类中定义了允许同时打开多个连接的个数, 定义了连接的状态有三种: CLOSED 表示关闭、START 表示连接打开、OPEN 表示连接已经建立。提供的公共函数有:

```
TCPServer( int MaxNumConn);
```

功能描述:

此为 class TCPServer 的构造函数, 对各类变量进行初始化。定义了同时打开的连接数, 由所带参数 MaxNumConn 来定, 其大小不超过 MAXNCONN。并将各个连接的初始状态置为 CLOSED。由于网络的资源有限, 总共提供的连接数为 10 个, 须同时供以太网和 PPP 连接使用, 所以定义的以太网连接数目不能太多, 即 MAXNCONN 定义的值最多不超过 9, 一个提供 PPP 连接。在本应用案例中, MAXNCONN 定义为 6, 即最多同时打开 6 个以太网连接。

```
int LoadDatPro( int(*DatPro)(void* pMssgIn, int DatLength, void* pMssgOut));
```

功能描述:

加载一个数据处理的函数指针。数据处理的函数 DatPro() 为一个外部函数, 用户根据不同的需求可自行定义适合的数据处理函数, 通过 LoadDatPro() 可将此函数加载到 class TCPServer 内部中, 实现将类中定义的内部函数 DataProcessing() 指向外部函数 DatPro(), 调用 DataProcessing() 将执行外部的数据处理函数 DatPro()。

```
int ServerRunning();
```

功能描述: 作为 TCPServer 服务器主要运行函数。包括三项处理:

- 处理已建立 (OPEN) 的连接。首先检查各个连接的状态是否为 OPEN, 如果为 OPEN 状态, 将检查是否有数据可读, 如果有就读取数据, 然后进行数据处理 DataProcessing(), 实际上调用的是外部函数 DatPro(), 本应用案例提供的外部数据处理函数 DatPro() 执行的是将以太网接收的数据作为 EMAIL 的内容通过 PPP 客户端发送出去, 发送邮件的结果 (成功或失败) 以串的形式返回到以太网。最后再检查是否有数据可发送, 如果有将把数据发送到 TCP 客户端。
- 检查已经打开的连接是否建立连接。检查连接的状态是否为 START, 如果为 START, 再检查该连接是否建立。
- 打开一个新的连接。首先检查各个连接的状态, 如果没有一个连接处于 START 状态, 并且至少有一个连接处于 CLOSED 状态, 将启动打开一个新的连接, 如

果打开成功，将状态置为 START。否则的话将不能打开新的连接。

```
int DataProcessing(void* pMssgIn, int DatLength, void* pMssgOut);
```

功能描述：

外部定义的处理函数，其中 void 指针 pMssgIn 指向输入参数缓冲区，DatLength 为输入参数的长度，单位为字节；void 指针 pMssgOut 指向输出数据缓冲区，返回值表示数据缓冲区的长度。返回值大于 0 表示正常返回；小于 0 表示有错，其值为出错代码。应用处理程序可通过 2 个 void 指针来分别定义相关的输入输出数据结构。

```
int CloseAll( );
```

功能描述：关闭所有已打开的连接。

上述的 TCPserver 仅作为用户在开发自己的应用程序的一个参考，采用 C++ 的类来设计程序有助于得到更加稳定的程序结构。TCPserver 类的具体操作将在第六部分结合其它程序共同介绍。

FTP 服务器作为 TCP 上的一种典型应用服务器，在许多场合都能找到其应用。在本应用方案中包含了 FTP 服务器接口函数如下：

```
int FTP_server( void* pTsk )
```

功能描述：

FTP 服务器运行程序，FTP 的实现是以非阻塞方式和阻塞方式相结合为特征的，当未与客户端建立连接时，处于非阻塞方式，函数可返回；当与客户端建立连接后，FTP 服务器转入阻塞模式，这时需等待 FTP 推出后，函数才会退出。

输入参数：void* pTsk。pTsk 指向下面定义的一个数据结构：

```
struct MtaskHandler
{
    int tid;
    int priority;
    int status;
    void (*Reload)();
};
```

```
};
```

pTsk 主要为在多任务操作系统使用时提供必要的参数。实现 FTP 任务与其他任务的同步和通讯。如果系统启动 Watchdog，函数指针 Reload 指向相应的 Watchdog 加载函数，这样可保证在 FTP 操作过程中也能对 Watchdog 进行正常加载。在单任务操作系统，如果未使用 Watchdog，可把 pTsk 设为 NULL。

返回值	描 述
0	等待 FTP 连接打开
1	等待 FTP 连接建立
2	一次 FTP 连接完成，可再次调用该函数

第三部分 通过 PPP 访问互联网

英创嵌入式网络模块带有 2 个标准的异步串行接口 COM1 和 COM2，其中的 COM1 为 9 线制 RS232C，可方便地与有线 Modem 或无线 GPRS 模块实现对连。原因程序通过拨号首先建立点对点网络的物理连接，再调用配套的 PPP 协议库，英创的 PPP 协议库实现标准的 PPP 协商过程，包括链路配置、客户端认证、网络配置三个子进程，最终建立 PPP 的连接。一旦 PPP 链路建立，就意味着应用程序所在的客户端已从 PPP 的另一端获得了 IP 地址，已可通过 PPP 访问互联网了。本部分将就如何访问互联网及互联网上的相关应用作一说明，相关应用包括 FTP 获取文件及收发邮件。

在本方案中的 PPP 接口函数的一个重要特点是它们与具体的物理网络无关，用户可在零 Modem (null-modem)、标准拨号有线 Modem、GPRS 无线模块三种不同的配置下使用 PPP 接口函数。点对点物理网络连接的建立在第四、五部分介绍。

PPP 协议接口函数说明

PPP 协议库主要包括了 PPP 验证、连接过程。提供的函数包括设置用户名称、设置用户密码、启动 PPP 连接、检查 PPP 状态等操作。库函数主要是针对 PPP 的客户端的应用，所有函数的调用都应在网络初始化成功之后。下面就各个函数作一说明。

`int PPPSetupLink ();`

功能描述：作为 PPP 的客户端，启动 PPP 连接，进行 PPP 认证，并检查 PPP 状态。

返回值	描 述
0	PPP 打开、PPP 认证成功，进入网络层协议状态。
<0	PPP 打开、PPP 认证失败。

`int PPPGetState ();`

功能描述：获取 PPP 状态

返回值	描 述
1	PPP 打开、PPP 认证成功，进入网络层协议状态。

0	PPP 处于协商状态
-1	PPP 处于关闭状态，可再重新建立 PPP 连接

`int PPPSetUsername (char* NewName);`

功能描述：设置新的用户名称。

输入参数：char* NewName

需要设置的新的用户名称，如果 NewName 为 NULL，则沿用缺省的用户名，“guest”

备注：

PPP 库只保留 NewName 的指针，因此用户应把 NewName 作为全局或静态量，切忌作为自动变量放在堆栈里。

`int PPPSetPassword (char* NewWord);`

功能描述：设置新的用户密码。

输入参数：

char* NewWord 需要设置的新的用户密码，如果 NewWord 为 NULL，则沿用缺省的用户密码，“888”

备注：

PPP 库只保留 NewWord 的指针，因此用户应把 NewWord 作为全局或静态量，切忌作为自动变量放在堆栈里。

`int PPPForceLinkDown ();`

功能描述：强制执行 PPP 终止操作，使 PPP 返回关闭状态。

`int PPPDisable (void interrupt (*modemhandler) (...));`

功能描述：暂停 PPP 操作，用外部 Modem（有线或无线）的中断函数替代当前中断函数，使外部 Modem 的接口函数直接操作嵌入式模块的串口。

输入参数：

void interrupt (*modemhandler) (...) 需要替换的 Modem 中断函数的函数指针。

```
int PPPEnable ();
```

功能描述：恢复缺省的串口中断函数并重新启动 PPP 操作。

备注：

PPPSDisable (...) 和 PPPEnable () 应成对使用。对通过 Modem 上网的应用，在网络初始化完成后，可首先调用 PPPDisable (...) 关闭 PPP，进行有关的拨号操作，建立物理连接后，再调用 PPPEnable () 打开 PPP。

应用案例及其实现

在第二部分中我们已介绍了以太网以服务器方式同时打开的多个 TCP 连接，响应本地局域网的数据请求的实现方法。这一部分我们着重介绍 PPP 建立过程、通过 PPP 实现 FTP 文件传送及电子邮件收发。

建立 PPP 链接

这里介绍的访问互联网是通过有线 Modem 实现的，有线 Modem 必须接在英创嵌入式模块的串口 COM1 上，因为与其配套的 PPP 协议库只支持接在串口 COM1 的操作。具体实现的方法包含在应用案例的主函数中，模块 PPPROUT.CPP 包含了主函数 main()。

首先调用 InitEthernetNet(char*IPStr) 初始化网络接口，包括以太网接口和 PPP 接口。带上的 IP 地址参数只对以太网接口有意义，对于基于 COM1 的 PPP 接口是不需要设定 IP 的。

然后进行 Modem 拨号，具体 Modem 拨号是如何实现的将在后一部分“标准 Modem 拨号的实现”中作详细说明。Modem 拨号实现中接收和发送数据是靠 COM1 的中断服务程序来实现的，所以在进行拨号之前，需将该中断函数替换 PPP 协议库的对应中断函数，调用函数 PPPDisable(COM_ISR); 在拨号成功之后，又需要将原来的中断函数替换回去，调用函数 PPP Enable()。到了这时，如果拨号（案例中拨号“163”）成功，已经在 PPP 的两端建立了一条物理连接。

PPP 物理上连通以后，调用函数 PPPSetUsername(“163”) 及 PPPSetPassword(“163”) 分别设置 PPP 连接的用户名称和密码。然后调用 PPPSetupLink() 作为 PPP 的客户端，启动 PPP 连接，进行 PPP 认证，其返回值为当前 PPP 连接的状态。如果返回 1 则表示认证通过，已经通过拨号“163”进入互联网。

互联网的应用

本应用案例中的互联网的应用包括了 FTP 文件传送和邮件收发。我们的协议库中使用了两种邮件协议：SMTP 和 POP 协议。SMTP 协议是一种基于 TCP 的应用协议，用于发送邮件。而 POP 协议支持从主机接收邮件。在互联网上一般需要对域名进行解析，所以还提供了域名解析函数。具体的应用函数定义如下：

```
int DNSresolve( char* fullname, char* IPStr )
```

功能描述：解析域名，将域名转换为 IP 地址。

输入参数	描 述
char* fullname	域名地址（如“www.yahoo.com”）

输出参数	描 述
Char* IPStr	转换后对应的 IP 地址

返回值	描 述
>=0	转换成功
-1	不能从 DNS 服务器中获得 IP 地址
ENOBUFS (-15)	无数据缓冲区

```
int FTP_getput( char *host, char *file , int mode, char *FTPdir,
               char *FTPusername, char *FTPpassword )
```

功能描述：

以 FTP 客户端方式向远端 FTP 服务器发送文件或从远端 FTP 服务器获取文件。

输入参数	描 述
char *host	远端主机的 IP 地址，如“192.168.201.34”
char *file	被操作的文件名，如“myfile.txt”
int mode	0: 从远端服务器获取 ASCII 文件 1: 从远端服务器获取 2 进制文件

	2: 向远端服务器发送 ASCII 文件 3: 向远端服务器发送 2 进制文件
char *FTPdir	所操作的文件在 FTP 服务器上的存储路径, NULL 表示当前目录
char *FTPusername	登录时使用的用户名, 如 “guest”
char *FTPpassword	登录时使用的密码, 如 “888”

返回值	描 述
0	该函数调用成功
! =0	该函数调用失败

通过 FTP 服务器中转来实现点对点的数据交换是利用互联网进行数据通讯的最简便的方法。客户可通过申请代理服务器的方法来获得廉价而稳定的 FTP 服务器, 通过 GPRS 的文件传输效率也很高, 文件内容及格式均由用户自己定义, 这对应用程序的设计是很方便的。有关利用 FTP 文件传送机制, 在互联网上实现点对点数据通讯的问题请参见相关的技术方案或向英创公司技术服务部咨询。

```
int SetMail (struct MailPara* pMP);
```

功能描述: 设置邮箱有关参数, 参数由 struct MailPara 定义如下:

```
struct MailPara
{
    char DNSdomain[40];    // 邮箱的域名地址
    char smpt_server[40]; // 邮件发送服务器
    char pop_server[40];  // 邮件接收服务器
    char username[20];    // 邮件用户名
    char password[20];    // 邮件用户密码
};
```

返回值	描 述
-----	-----

0	邮件发送成功
-1	无法识别自身域名
ENOBUFFS (-15)	无数据缓冲区
EHOSTUNREACH (-11)	无法找到邮件服务器

int SendMail (char* to, char* subject, char* mailbody)

功能描述： 发送一个邮件。

输入参数	描 述
Char* to	目的 email 地址
Char* subject	发送邮件的主题，必须是 ASCII 码串
Char* mailbody	发送邮件的内容，必须是 ASCII 码串

返回值	描 述
>=0	邮件发送成功
-1	无法识别自身域名
-2	与邮件服务器协商失败

备注：本函数在调用前须先调用 SetMail (struct MailPara* pMP) 对本邮箱进行设置。

int GetMail (char* from, char* subject, char* mailbody)

功能描述：从 POP 服务器接收邮件。可反复调用以获得多个邮件。

返回值或返回参数	描 述
0	邮件接收正常，但无新邮件
1	接收到新邮件。
-2	与邮件服务器协商失败
Char* from	Email 的发送地址
Char* subject	接收邮件的主题，必须是 ASCII 码串
Char* mailbody	接收邮件的内容，必须是 ASCII 码串

备注:

本函数将丢弃可能的邮件附件。

本函数在调用前须先调用 `SetMail (struct MailPara* pMP)` 对本邮箱进行设置。

电子邮件收发是互联网上最常用的通讯手段，目前许多手机已能实现邮件的收发。一般来说，电子邮件的传送，需要把传送的内容转换成 ASCII 码。目前在 PC 上收发的邮件均是按照多信息文本方式进行编码的，若利用英创嵌入式网络模块接收这样的邮件，就需要对文件内容进行解码。简单的办法是采用纯文本方式发送邮件，这样接收到的邮件内容就是简单的 ASCII 码串，用户应用程序只需对 ASCII 码串进行必要解析，就可获得相应的信息。从英创嵌入式网络模块发出的邮件是采用纯文本方式编码的。相比 FTP 文件传送方式，采用电子邮件方式进行数据传送的效率要低一些。

第四部分 标准 Modem 拨号的实现

本部分主要介绍通过标准的 AT 指令来实现标准 Modem 的拨号功能。应用程序执行 AT 命令就是通过串口把 AT 指令代码串发送到 Modem，当 Modem 接收到一条 AT 命令之后都会向串口发送结果码以对接收到的命令作出响应，应用程序通过串口读取返回码，并作相应的处理，就可判断本次操作是否成功。

提供的函数包括对串口读写操作的中断服务程序，以及 Modem 拨号呼叫、Modem 应答呼叫、强制挂断 Modem 连接、读取 Modem 状态等函数，这部分的函数，我们以源代码的形式提供给用户，我们提供的是一个最基本的 Modem 操作，用户可以直接采用我们提供的函数，进行拨号上网，也可以针对不同的 Modem，修改执行的 AT 命令，以更好地发挥 Modem 的作用。

中断程序支持串口收发，通过内部的缓冲区与应用程序交换数据。当应用程序需发送数据时，首先把数据写入发送缓冲区，再启动硬件发送，中断服务程序将把发送缓冲区的数据逐一发送直至发送缓冲区为空；当串口接收到数据时，将触发中断服务程序把数据填入接收缓冲区，应用程序只需直接读取缓冲区的数据即可。下面就各个函数作一说明。

```
void interrupt COM_ISR( __CPPARGS );
```

功能描述：串口接收、发送数据的中断响应函数。

```
int Modem_DialOut( char* phonenumber );
```

功能描述：Modem 拨号呼叫，返回 0 表示拨号连接成功，返回-1 表示拨号失败。

```
int Modem_DialIn( );
```

功能描述：Modem 应答呼叫，返回 0 表示 Modem 连接建立成功，返回-1 表示连接失败

```
void Modem_DialDown();
```

功能描述：强制挂断 Modem 连接，使 Modem 处于离线状态。

```
int Modem_Status( );
```

功能描述：

检查 Modem 状态。返回 0 表示 Modem 处于在线状态。-1 表示 Modem 处于离线状态。

第五部分 通过 GPRS 模块进入互联网

在本 GPRS 应用案例中，采用的是 SIMENS 的 MC35 无线网络模块，通过 GPRS 模块进入互联网的方法和有线 MODEM 拨号上网的方法很类似，也是通过串口将有关的 AT 指令代码串发送到无线网络模块上，无线网络模块会对接收到的指令作出响应，并向串口发回结果码。

和有线 MODEM 一样，MC35 也有 AT 指令和数据通讯两种工作模式。当 MC35 加电后，应用程序需通过 IO 口（如 ETR232E P4 口的 D7 位）操作 MC35 的 IGT#控制位，使 MC35 正式启动，启动过程大约 3-5 秒，若 MC35 接有有效的 SIM 卡，MC35 将附着在 GPRS 网络。与对有线 MODEM 的操作类似，应用程序对 MC35 的串口读写操作仍然是由中断服务程序来实现，在进行 AT 指令操作前，同样需运行 PPPDisable (...); 待 MC35 进入数据通讯状态后，再运行 PPPEnable (...).

本方案通过提供的 int GPRS_Logon () 函数来设置 MC35，使其进入数据通讯状态，函数返回 0 表示 MC35 成功进入数据通讯状态，应用程序可启动 PPP 配置过程，最终建立 IP 网络连接；函数返回-1 表示 MC35 模块未能附着于 GPRS 网络；函数返回-2 表示网络登录失败。在函数 GPRS_Logon () 中调用的是标准的 SIEMENS 的 AT 指令集，下面就该函数中调用的 AT 指令作一介绍。

“AT+CGATT?” 该指令用于查询 MC35 是否已附着在中国移动的 GPRS 网络，MC35 将返回当前状态。

“AT+CGATT=1”该指令用于设置 MC35 附着于中国移动的 GPRS 网络，操作成功 MC35 将返回 OK。

“AT+CGDCONT=1,”IP”,”CMNET” ”该指令用于设置中国移动的 GPRS 节点服务器的名称和属性，操作成功则返回 OK。

“ATD*99***1#”该指令用于拨通连接中国移动的 GPRS 节点服务器，操作成功则返回 CONNECT。这时 MC35 已进入 PPP 模式，不在响应其他 AT 指令，若应用程序在一段时间内不对 MC35 进行有关 PPP 操作，MC35 将返回 AT 指令模式。

在无线模块成功连入中国移动的 GPRS 节点服务器后，应用程序就可调用 SetupPPPLink () 函数进行后一步的 PPP 连接。PPP 连接、认证通过以后，即应用程序就已通过无线模块成功进入了互联网。

第六部分 通过无线模块收发短消息

在本应用案例中，采用的是 SIMENS 的 MC35 无线网络模块，通过 GSM 进行短消息的收发，和 GPRS 上网一样其操作的方法也是通过串口将有关的 AT 指令代码串发送到无线网络模块上，无线网络模块会对接收到的指令作出响应，并向串口发回结果码。

应用程序对 MC35 的串口读写操作仍然是由中断服务程序来实现，在进行 AT 指令操作前，同样需运行 PPPDisable (...), 以保持与其他应用程序一致。本方案通过提供的 int SendSMS(char* Number, char * Content) 函数来发送短消息，函数返回 0 表示发送成功，函数所带参数其中 char* Number 为发送的目的手机号，char * Content 为需发送的短消息内容；int ReceiveSMS(char* Content)函数用于接收短消息，函数返回值大于 0 表示有短消息，并接收成功，接收的数据放在 Content 串中，函数返回值为 0 表示没有短消息，返回值小于 0 表示接收操作失败。Content 的最大长度为 140 个字节。在这两个函数中均调用的是标准的 SIEMENS 的 AT 指令集，下面就该函数中调用的 AT 指令作一介绍

由于目前中国移动还不支持通过 GPRS 进行短消息的收发，因此如果无线模块还处于 GPRS 附着状态，需发送指令“AT+CGATT=0”，该指令用于设置 MC35 脱离中国移动的 GPRS 网络，操作成功 MC35 将返回 OK。

“ATI”该指令用于读取无线模块 MC35 的版本信息，操作成功 MC35 返回 OK。

“AT+CMFG=1”该指令用于选择短消息的格式，选择 1 为文本格式，选择 0 为 PDU 格式，操作成功 MC35 返回 OK。本方案中均采用文本格式。

“AT+CMGS=1390800****”该指令用于发送短消息，操作成功 MC35 返回 OK

“AT+CMGR=1”该指令用于接收短消息，操作成功 MC35 返回 OK。

“AT+CMGD=1”该指令用于删除短消息，操作成功 MC35 返回 OK。

第七部分 应用程序设计说明

本案例提供了三个完整的应用程序，一个应用程序利用有线 MODEM 拨号上网，进行邮件收发，应用程序从以太网接收 ASCII 命令数据串，通过 PPP 连接互联网上的指定邮件服务器，把从以太网接收的数据作为 email 内容发送出去或接收邮件服务器中的 email，并把邮件收发结果信息返回以太网相应的客户端。第二个应用程序与上一个程序类似，不同在于采用 GPRS 无线模块连接到互联网。第三个应用程序利用无线模块实现了短消息的收发，应用程序从以太网接收 ASCII 命令数据串，并将接收的数据作为短消息内容发送到指定的目的手机上，或接收发送到该模块的短消息内容，并把短消息收发的结果信息返回以太网相应的客户端。

应用方案以运行库的形式 (ENETPPP.LIB) 向用户提供可调用的 C 函数壳层。应用方案总体以 BC/C++集成开发环境 (IDE) 中的工程文件形式提供。与 PRJ 工程文件和运行库一起构成完整应用方案的文件还包括：头文件及相关的应用模块。

相对于运行库头文件有：ETR_TCP.H、ETR_PPP.H 及 ETR_APP.H，其中 ETR_TCP.H 包含了第二部分中定义的 TCP/IP 通讯接口函数，ETR_PPP.H 包含了第三部分中定义的 PPP 协议接口函数，ETR_APP.H 包含了第三部分定义在 TCP 之上运行的各个应用函数。

应用程序共提供了三个工程文件：

- (1) PPPROUT1.PRJ 有线 MODEM 应用程序
- (2) PPPROUT2.PRJ 无线 GPRS 应用程序
- (3) PPPROUT3.PRJ 无线模块短消息收发应用程序

应用模块包含有：

- (1) MODEM.CPP：主要实现对标准有线 Modem 的操作。
- (2) GPRS.CPP：主要实现对无线网络模块 MC35 的操作。
- (3) TCPSERV.CPP：实现了以太网的通讯，以服务器方式同时打开多个 TCP 连接，响应本地局域网的数据请求。
- (4) PPPROUT.CPP：包含了主函数 main ()，应用主程序采用的是单任务、大循环、不断查询的方式编写。为简化程序流程，采用一个请求处理完毕后，再接受下一个请求的数据处理原则。PPPROUT.CPP 另一个重要函数是数据处理函数 GetResponseFromPPP (...)，注意数据处理的原型一定要符合第二部分应用案例

中对其的定义，否则可能出现不可预料的结果。

利用 FTP 文件传送的功能，可对 GPRS 数据传送的实际情况作一测试，其结果如下：

操 作	测试硬件环境	实际数据量	GPRS 记录数据量	传送数据率
上 传	NetBox+MC35	90.8Kbytes	98Kbytes	1Kbytes/s
下 传	NetBox+MC35	90.8Kbytes	98Kbytes	2Kbytes/s