# 功能需求

## 1、DNS支持

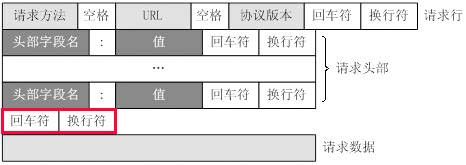
只需要知道Web server的域名，就可以通过linux的系统调用gethostbyname直接获得。前提是系统中已经正确地配置了DNS，或者说能够从路由器动态获取到DNS。而且Web server的域名是在电信运营商的域名解析服务器中登记。

## 2、DHCP支持

原有嵌入式linux是静态配置IP的，需要加入DHCP功能。BUSYBOX内自带的DHCP client功能，广为采用，容易移植。

## 3、HTTP方式与后台通讯

HTTP的报文相对比较简单，而且只是使用post方式。因此不选用开源的HTTP库（如libcurl）。因为这些库支持的功能比较全，但我们不需要。同时移植这些库也有一定的难度。确定采用的方案是自己组装HTTP报文。



如：

POST <http://q.sto.cn/track.aspx>? HTTP/1.1 \r\n

Content-Type: application/json\r\n

CharacterSet:utf-8\r\n

Content-Length:15\r\n

\r\n

wen=35646456456

## 4、SSL加密

HTTP报文在发出去之前，需要使用SSL加密。收到的HTTP报文也是经SSL解密的。SSL库将使用openssl，这是使用最广泛的开源ssl实现，支持规范要求的SSLv2，SSLv3，X.509等。

openssl在已有的嵌入式linux中未支持，所以需要移植到文件系统中。移植过程有一定的难度，而且需要熟悉openssl的接口。

## 5、JSON数据格式

HTTP报文中的Body部分使用JSON数据格式，即指Request和Response部分。

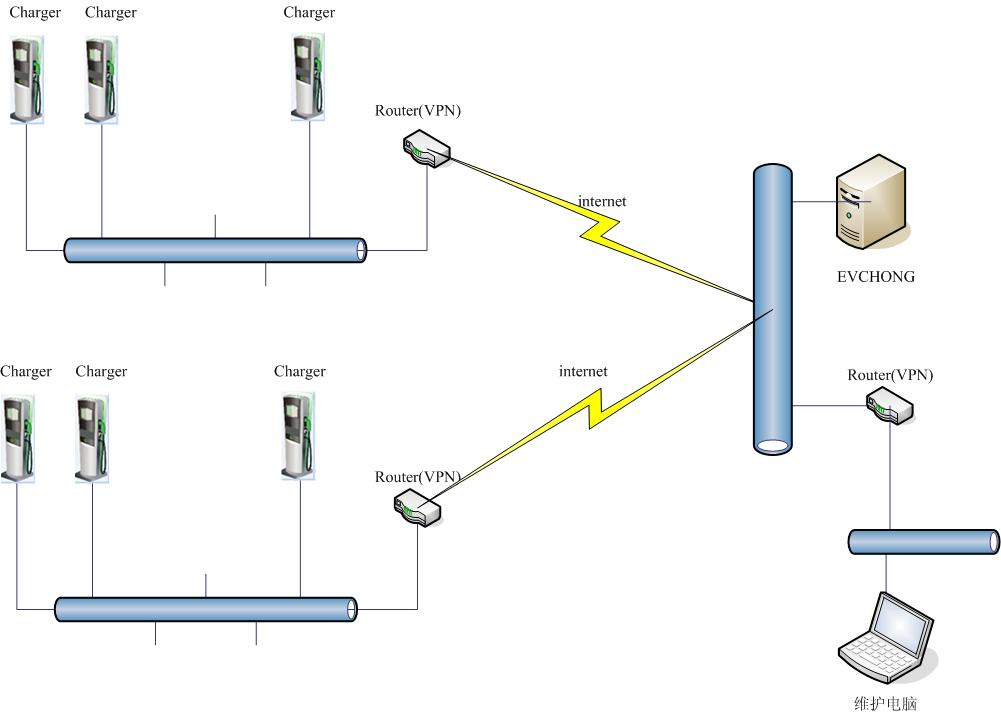
<http://sourceforge.net/p/cjson/code/HEAD/tree/>

## 6、UTF-8编码

指定HTTP头部中的属性charset为UTF-8，则HTTP报文中的Body部分使用UTF-8编码。

# 系统架构

数据流图



网络拓扑图

# HTTP层接口

HTTP层初步定义接口如下：

/\*\*

\* @brief HTTP POST请求

\* @param strUrl 输入参数,请求的Url地址,如:http://www.baidu.com

\* @param strRequest输入参数,使用JSON格式的请求数据

\* @param strResponse 输出参数,返回的响应内容（已经去HTTP头部）

\* @return 返回是否Post成功

\*/

int HttpsPost(const std::string &strUrl, const std::string &strRequest, std::string &strResponse);

在接口内部需要从URL中提取出域名，通过DNS获取IP；创建socket，组装HTTP包，经加密后发送到Web Server，再循环接收数据，期间可能会阻塞。收到数据后，再close socket。