

# WH-NB73 接入透传云说明

文件版本：V1.0.3



## 目录

WH-NB73 接入透传云说明.....	1
1. 透传云介绍.....	3
1.1. 透传云是什么? .....	3
1.2. 透传云特点.....	3
1.3. NB 模块为什么和透传云更配? .....	4
2. 透传云接入演示.....	4
2.1. 透传云账号注册.....	4
2.2. 添加设备.....	6
2.3. 数据透传演示.....	8
2.4. 0 编程远程监控演示—温度计 .....	11
2.5. 有人扩展 Modbus.....	16
3. 联系方式.....	17
4. 免责声明.....	17
5. 更新历史.....	17

# 1. 透传云介绍

## 1.1. 透传云是什么？



有人透传云 (<http://cloud.usr.cn/>) 主要是为解决设备与设备、设备与上位机 (Android、IOS、PC) 之间远程相互通信而开放的平台。

透传云专为工业应用设计，主要解决了软件如何与 NB-IoT/GPRS/3G/4G DTU、串口服务器进行远程数据交互的问题，从而实现工业物联网的数据采集和远程控制。

## 1.2. 透传云特点

云组态：0 编程快速实现远程监控项目，只要简单设置，服务器自己完成设备对接和协议解析。详见：

[http://cloud.usr.cn/simple\\_instruction.html](http://cloud.usr.cn/simple_instruction.html)

**云接口：**利用二次开发接口实现自定义项目。详见：[http://cloud.usr.cn/development\\_instruction.html](http://cloud.usr.cn/development_instruction.html)

**云交换机：**支持远程数据透传功能，利用二次开发接口或虚拟串口软件中转原始数据流。

**多种协议：**接入协议支持 MQTT、CoAP、TCP、LoRaWAN 等，上传数据支持 Modbus RTU/TCP 协议。

## 1.3. NB 模块为什么和透传云更配？

专为低功耗而生的 CoAP 协议，众所周知 NB-IoT 最突出的特点就是低功耗，为了实现低功耗功能增加了休眠模式，导致模块不能主动收到服务器数据，并且 NB 模块目前只支持 UDP 协议，那么为了更好的实现数据传输，我们在模块内部专门内嵌了 CoAP 协议，可以帮助用户只需要简单的一条发送指令就可以百分之百的将数据发送到服务器。

有人透传云支持 CoAP 协议接入，用户只需要在透传云进行设备的添加，就可以实现将用户发送的数据发送到透传云服务器上，内嵌的 CoAP 机制可以保证数据的准确性，用户可以基于透传云的二次开发 SDK 读取到用户的数据和向用户设备发送指定的数据。

透传云更支持 Modbus 协议，用户是需要进行简单的配置可以实现自己的应用程序，做到 0 编程快速实现远程监控。同时有人透传云后续会推出更多功能，方便客户快速实现自己的应用，敬请期待！

目前电信运营商不允许将数据发往私有 IP，所以使用电信运营商的客户只能访问电信云或者华为云平台，而这两个平台入门难度高，设计复杂，导致客户在接入时存在困难，有人透传云简单的授权和配置使用方式方便用户快速上线项目。

## 2. 透传云接入演示

### 2.1. 透传云账号注册

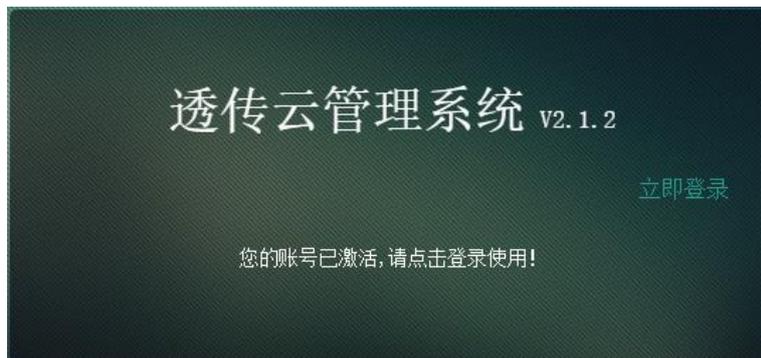
(1) 使用浏览器访问 <http://cloud.usr.cn/>，并点击右上角控制台按钮跳转到登录界面。



(2) 点击注册按钮，填写相关注册信息后点击注册。



(3) 点击注册后，邮箱会收到一封验证邮件，点击验证。



(4) 使用注册的账号进行登录，登录成功后显示下面界面：



## 2.2. 添加设备

(1) 首先登录到控制台，进入到设备管理当中的设备添加一栏。



(2) 添加新设备需要填写以下信息：

所属用户和设备分组：默认即可；

设备名称：用户自定义，建议按照一定规律命名；

设备类型：勾选“电信 CoAP/NB-IoT”；

IMEI 和 SN：填写设备贴膜上的 IMEI 和 SN 码；

**注：SN 和 IMEI 长度均为 15 位，输入 SN 码时不要携带“SN”字样，如果校验出错请将 SN 和 IMEI**

提交提交工单 (<http://h.usr.cn/>) 寻求技术支持解决。

通信协议：根据客户需求填写，本文档以“数据透传”为例。

所属用户\*：

设备分组\*：

设备名称\*：

设备类型\*： 默认设备  LoRaWAN模块  网络IO  二维码添加  LoRa模块  
 LoRa集中器  CoAP/NB-IoT  电信CoAP/NB-IoT

IMEI\*：

SN\*：

通讯协议： Modbus RTU  数据透传

(3) 接下来可以上传设备的自定义图标以及模块的位置信息。

设备图片：  
仅支持jpg、gif、png格式；大小为1M以内

地图位置：  
山东省济南市历下区  
经度：117.02496707, 纬度：36.68278473

(4) 设置完成后点击“保存”完成设备的添加，添加后的设备可在设备列表里面查看和管理。

状态	名称/位置	设备编号	所属分组	设备类型	通讯协议	采集频率	所属用户	修改时间	操作
☐	NB73-1 济南市高新区奥盛大厦	356566075158303	默认分组	电信CoAP/NB-IoT	数据透传	--	wangyawei	10:11:03 2018-03-13	🔍 📄 🗑️ ⚙️ ☰

## 2.3. 数据透传演示

- (1) 测试前看下整个框图，接下来演示就是围绕下面的框图进行；



- (2) 参考 2.2 章节添加一个设备，并选择通讯协议为数据透传；
- (3) 在透传云左侧标题栏中找到“二次开发”并进入界面，选择一种二次开发 Demo 进行测试；

设备接入

手机端查看

应用开发 >

更新日志

二次开发

SDK包：

名称	SDK形式	使用文档	Demo下载
PC端开发	DLL(下载地址)	<a href="#">查看</a>	(C#) (Delphi) (LabView)
Web网页端开发	Js包(下载地址)	<a href="#">查看</a>	(在线演示)
JAVA或安卓开发	Jar包(下载地址)	<a href="#">查看</a>	(APK) (源码)

demo下载

协议文档：数据服务器基于MQTT协议，透传云定义了Topic规则，上述SDK均遵守本协议，用户也可自行实现本协议来接入透传；

- (4) 这里我们使用最方便的网页版 demo 进行演示，点击在线演示，首先使用透传云账号进行登录；

登陆

用户名:  密码:

连接成功

- (5) 找到云交换机一栏，订阅一个设备，设备的 ID 就是添加设备时的 IMEI。

## 云交换机

订阅 & 取消订阅 (订阅后收到设备推送消息)

操作

单个设备  
设备ID:   
(多个设备用英文格式的","分隔)

账号下全部设备

单发 设备ID:

群发  16进制

已订阅设备

- 356566075158303

(6) 订阅完成后模块发送的数据将在云交换机的“收到消息”窗口显示。

收到消息

已接收: 0

(7) 模块参数设置, 首先将模块串口连接到电脑, 电脑端串口调试助手使用 9600,N,8,1 配置后打开串口, 给模块上电, 收到开机信息: “[WH-NB73]”

[WH-NB73]

(8) 设置透传云地址, 使用 AT 指令将设置透传云地址为: **117.60.157.137** 端口为 **5683**

AT+NCDP=117.60.157.137,5683

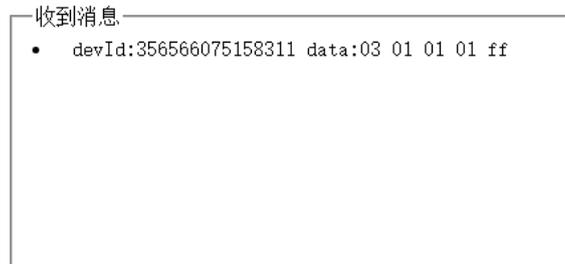
(9) 设置完成发送 AT+S 保存参数或发送 AT+NRB 重启模块模块, 重启后使用 AT+CGATT?指令查询驻网

状态，具体操作参考用户手册或者 AT 指令集。

(10) 发送一条数据到服务器，使用 AT+NMGS 指令发送 5 个字节长度的数据 03010101FF，确保指令返回的是 OK，如果 ERROR 就继续发送。



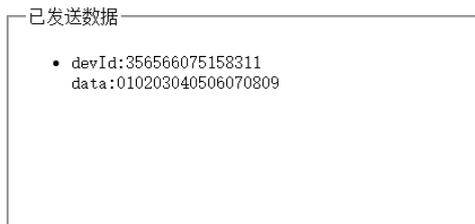
(11) 在演示 demo 当中查看数据接收情况，NB 模块发送数据延时较大，需要稍等一下才有显示。



(12) 用户发送数据到模块，由于 NB 设备平时处于休眠状态，用户数据无法直接下达，需要等用户在主动发送一次数据后，平台将缓存的数据发送到模块端。

(13) 测试前请参考用户手册或者 AT 指令集设置 AT+NNMI=2，只有设置该指令后收到消息会通知到串口。

(14) 演示 demo 发送一条数据，需要设备 ID（模块的 IMEI），输入数据，点击发送，注意的是模块如果当前在休眠模式中，该数据会缓存到云平台等待下次模块主动发送数据时再下发。



(15) 我们先发送一条数据到服务器，然后观察串口打印，我们会观察到串口打印一个+NNMI 的信息，说明当前有一条数据接收到，数据已经存到缓存里。接下来使用 AT+NMGR 指令读取改数据即可。

```

OK
OK
OK
+SDMT 数据到达通知
9,010203040506070809
OK
读取到的数据

读取数据指令
226
AT+MMGR
    
```

- (16) 到此为止整个的数据透传的演示结束, 涉及的 AT 指令的内容请参考用户手册及 AT 指令集获取更多的信息;
- (17) 在用户正式使用中可以通过透传云的 API 接口获取到设备上传至服务器中的数据, 涉及的透传云的二次开发信息请参考透传云二次开发帮助信息: [http://cloud.usr.cn/development\\_instruction.html](http://cloud.usr.cn/development_instruction.html)

## 2.4. 0 编程远程监控演示—温度计

本次演示主要实现以下内容: 用户有一个温度采集的传感器, 按照 Modbus 协议通过 NB 模块上传温度信息, 在平台上展示该传感器的信息, 并添加一个报警值, 当值超过设置温度时通过微信推送报警信息。

- (1) 在添加一个设备前, 首先我们添加一个数据模板, 云服务将根据数据模板解析协议, 相同格式的设备可关联同一个模板。请先添加数据模板, 然后添加数据点, 最后添加设备。



数据模版 首页 / 数据模版

添加
批量删除

<input type="checkbox"/> 全选	模版名称	关联数据点数	更新时间	操作
<input type="checkbox"/>	温度计	0	2018-01-23 18:36:04	<span style="font-size: 1.2em;">✎</span> <span style="font-size: 1.2em;">✖</span> <span style="border: 1px solid red; padding: 0 2px;">+</span> <span style="font-size: 1.2em;">☰</span>

1 / 共1条

添加数据

名称\*：

关联数据模版\*：▼ 1

数据类型\*： 数值型  开关型

寄存器\*：

数值类型：2字节无符号整数 ▼

读写\*： 只读  读写

单位：

公式：

变量描述：

提交

其中寄存器值将在最后 Modbus 协议描述中讲解。

(2) 然后添加一个设备，添加设备时选择设备通讯协议为 Modbus RTU，其他方式参考上面章节方式进行添加。关于 Modbus 协议的内容请参考最后一个章节。

需要关注的是从机一栏，关键信息为设备号和关联模板，此处的设备号相当于 Modbus 从设备地址，此处测试填写 1

通讯协议:  Modbus RTU  Modbus TCP  数据透传

采集频率: 主机(主动上传)

所属用户: wangyufeng

设备图片:  仅支持jpg、gif、png格式; 大小为1M以内

从机 ? : + 增加从机

序号	名称	设备号 ?	关联模版
1	温度计1	1	温 <input type="button" value="x"/> 温度计

(3) 设置完成后, 我们在设置报警信息, 选择触发器管理一栏, 选择添加触发器。

名称\*: 高温报警

选择数据模版\*: 温度计

选择数据点\*: 温度计

触发条件\*: 数值高于B

90

报警方式\*:  微信  邮件

保存

(4) 我们选择的是微信报警, 需要我们使用微信关注公共账号并进行账号的绑定, 微信扫描下面二维码关注平台。



扫描二维码关注公众平台，选择右下角“透传云”，选择“微信绑定”即可。

(5) 数据上传测试，根据上面设置的数据类型我们要上传的数据如下：01 46 00 00 00 01 02 00 64 22 AD

01 代表从设备地址

46 代表寄存器单元类型

0000 为起始地址

0001 为寄存器单元长度

02 位数据长度

0064 为数据

22AD 为 CRC16 校验，其中数据 0064 转化 10 进制就是 100。

使用 AT+NMGS=11,01460000000102006422AD 进行发送。

(6) 点击进入监控中心一栏，选择列表展示，可以看到当前数据点的值更新为了 100



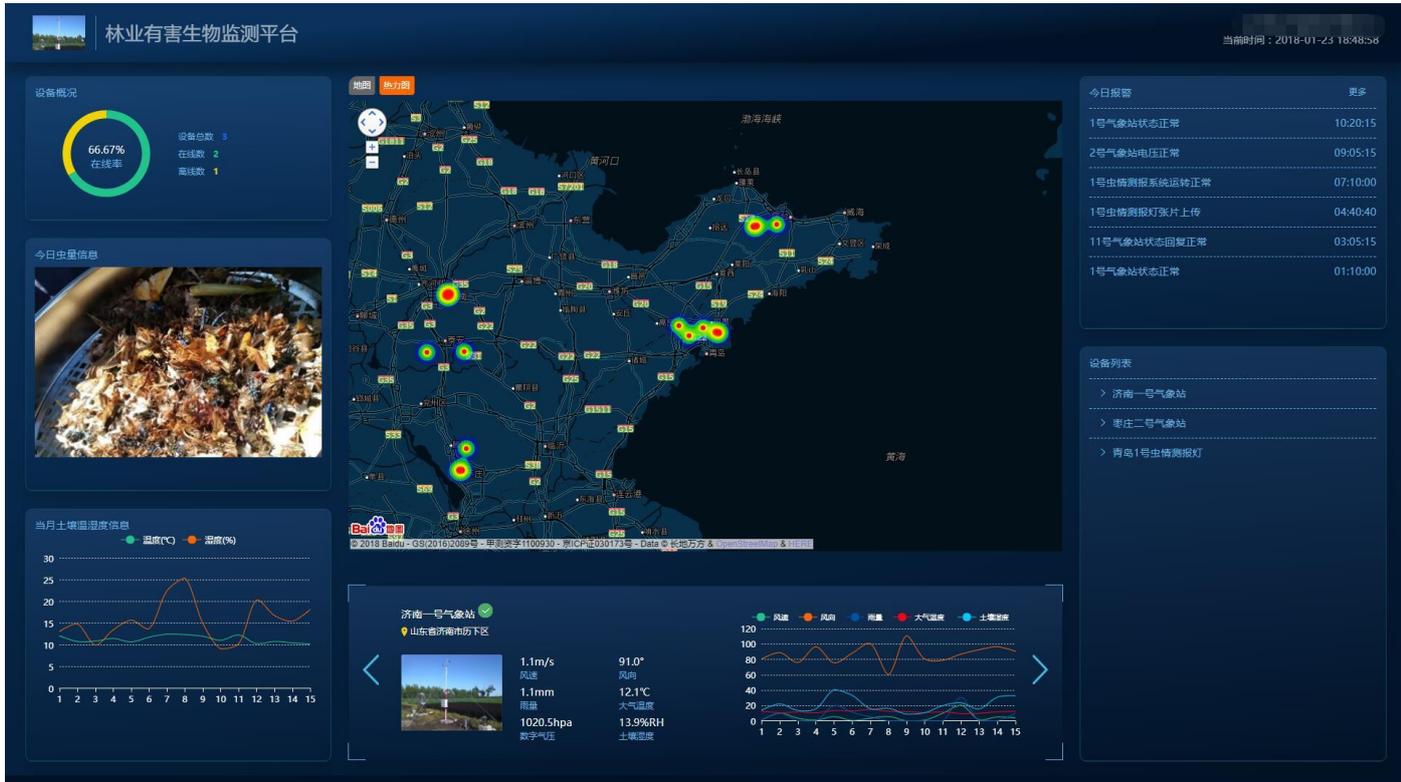
(7) 由于 100 大于设置的报警值 90，所以此时查看微信收到了来自透传云的报警推送。



(8) 到此为止，一个基于透传云的远程监控报警应用已经完成，真正的实现 0 编程，后续透传云将支持更

多的功能，敬请期待！

(1) 上面介绍的一个简单的演示，当然透传云的强大不止于此，甚至可以实现下面的应用：



## 2.5. 有人扩展 Modbus

有人扩展指令（本扩展指令非Modbus标准，仅适用于设备与透传云 协议通信的规则）														
为什么要做扩展指令	由于Modbus RTU协议适用于半双工的总线协议，只能一问一答，服务器做主机，接入设备做从机，可以实现主机主动采集和控制。但由于网络通讯考虑流量、服务器开发量、网络延迟等原因，无法实现类似在总线中毫秒级的轮询读取，而网络通讯属于全双工，能够同时双向对话，故扩展指令提出支持从机状态主动上报的规则。													
扩展协议应用场景	如Modbus协议的环境控制器，能够相应服务器做主机的采集和控制，但若前端人工主动控制，服务器必须等到下个轮询采集无法立即获取信息，利用扩展指令可实现环境控制器主动上报当前的状态。													
寄存器单元	功能码	作用	请求及响应格式											
线圈寄存器 CoilStatus (00001 - 0ffff) 1位 读写	0x45	上报多个线圈	示例：从0x0014开始写入10个状态值，数据内容为2个字节：十六进制CD 01（二进制 1100 1101 0000 0001）											
	(格式与写多个线圈相同)		请求：	设备号	功能码	寄存器起始地址	寄存器单元长度	字节数	数据				CRC校验	
				id	45	00 13	00 0A	02	CD 01	crcl	crch			
响应：	设备号	功能码	寄存器起始地址	寄存器单元长度	CRC校验									
			id	45	00 13	00 0A	crcl	crch						
离散寄存器 InputStatus (10001 - 1ffff) 1位 只读	0x42	上报多个离散寄存器	(格式与写多个线圈相同)											
保持寄存器 Holding Register (40001 - 4ffff) 16位 读写	0x46	上报多个保持寄存器	示例：将数据0x000A和0x0102写入从0x0002开始的两个寄存器单元											
	(格式与写多个保持寄存器相同)		请求：	设备号	功能码	寄存器起始地址	寄存器单元长度	字节数	数据				CRC校验	
				id	46	00 01	00 02	04	00 0A 01 02	crcl	crch			
响应：	设备号	功能码	寄存器起始地址	寄存器单元长度	CRC校验									
			id	46	00 01	00 02	crcl	crch						
输入寄存器 Input Register (30001 - 3ffff) 16位 只读	0x44	上报多个输入寄存器	(格式与写多个保持寄存器相同)											

协议解释：

- 1、寄存器起始地址从 0 开始，如：寄存器地址为 00 02，寄存器起始地址要减一写 00 01
- 2、数据、地址、长度等默认大端在前，但 CRC 校验是小端在前（低字节在前） crcl crch；
- 3、有人扩展协议仅适用于设备与透传云的网络通讯层；

### 3. 联系方式

公 司：上海稳恒电子科技有限公司

地 址：上海市闵行区秀文路 898 号西子国际五号楼 611 室

网 址：[www.mokuai.cn](http://www.mokuai.cn)

邮 箱：[sales@mokuai.cn](mailto:sales@mokuai.cn)

电 话：021-52960996 或者 021-52960879

使命：做芯片到产品的桥梁

愿景：全球有影响力的模块公司

价值观：信任 专注 创新

产品观：稳定的基础上追求高性价比

### 4. 免责声明

本文档提供有关 WH-NB73 产品的信息，本文档未授予任何知识产权的许可，并未以明示或暗示，或以禁止发言或其它方式授予任何知识产权许可。除在其产品的销售条款和条件声明的责任之外，我公司概不承担任何其它责任。并且，我公司对本产品的销售和/或使用不作任何明示或暗示的担保，包括对产品的特定用途适用性，适销性或对任何专利权，版权或其它知识产权的侵权责任等均不作担保。本公司可能随时对产品规格及产品描述做出修改，恕不另行通知。

### 5. 更新历史

2017-11-18 版本 V1.0.0 创立。

2017-11-28 版本 V1.0.1 增加 SN 和 IMEI 添加描述。

2018-01-25 版本 V1.0.2 增加有人 Modbus 描述