

ALIENTEK

**修订历史**

|  |  |  |
| --- | --- | --- |
| 版本 | 日期 | 原因 |
| V1.00 | 2018/6/15 | 首次发布 |
| V1.01 | 2018/6/20 | 增加NB卡与电信云平台账号流程简要描述 |



NBloT

工程技术笔记

目录

[1 概述 2](#_Toc16082)

[2 NBloT模组 3](#_Toc4781)

[2.1 移动NBloT模组 3](#_Toc17075)

[2.2 电信NBloT模组 3](#_Toc30097)

[2.3 联通NBloT模组 4](#_Toc793)

[2.4 NBloT模组选择 4](#_Toc23367)

[3 解决方案 5](#_Toc14042)

[4 现有同行 6](#_Toc1151)

[5 项目优劣 7](#_Toc5221)

[5.1 优势 7](#_Toc27744)

[5.2 劣势 7](#_Toc14043)

[6 项目实施 8](#_Toc25555)

[7 项目实施难点 9](#_Toc27808)

[7.1 政策难点 9](#_Toc17478)

[7.2 软件难点 9](#_Toc17313)

[7.3 硬件难点 9](#_Toc23903)

[8 NB卡与云平台账号申请 10](#_Toc2213)

[8.1 对接云平台账号申请 10](#_Toc7548)

[8.2 NB卡申请 14](#_Toc23148)

[8.3 服务热线 14](#_Toc13040)

[9 联系我们 15](#_Toc13918)

# 概述

Git 是一个分布式版本控制系统，而 GitLab 则用于管理 Git 服务器，它提供的主要功能有细致的权限控制以保证代码安全；通过合并请求提供好的代码回顾和增强合作；问题跟踪和 wiki。

在本文开始讲述之前，对文中可能出现的一些关键词语的意义作出解释，方便大家理解。

Git服务器： Git的服务器，方便各开发人员之间可以互相合作开发。

GitLab上的 Git服务器：GitLab上提供的Git服务器，每个项目都有一个独立的服务器，GitLab 基于这些服务器额外提供权限控制，合并请求等功能，用户也可以使用这些服务器来进行项目合作。

和 GitLab 协同作战：虽然项目成员之间可以使用GitLab上Git服务器自由交换数据，但是太过自由显然不利于项目的长久维护，对GitLab提供功能也没什么帮助。为了充分利用 GitLab 提供的功能，遵循一定的开发规范和流程是必须的。遵循这个规范和流程进行开发称为和 GitLab 协同工作。

本地：项目成员自己的电脑，是一个Git工作目录，它包含一个.git文件夹的本地版本库与检出的其它文件

# 安装软件

## Git软件安装

git for windows 是将 git 版本管理工具在 windows 上的实现，提供了和 Linux 一样的 Git命令行工具。在其官方主页上有 32 位安装版， 64 位安装版， 32 位便携版， 64 位便携版 4种版本，这里以 32 位便携版为例（之所以选择 32 位便携版，不仅因为其安装方便，而且因为 64 位版本因商密安全保护系统不能正常工作） 详述其安装过程

## TortoiseGit软件安装

安装了Git之后，就可以安装 TortoiseGit 了，在 windows 7 64bit 操作系统上推荐的安装版本是 TortoiseGit 2.4.0 64bit（与资源管理器集成的软件在 64 位系统上不能安装32 位软件）。开 始 安 装 前 ， 请 关 闭 其 它 所 有 程 序 （ 这 很 重 要 ）， 然 后 执 行 安 装 程 序TortoiseGit-2.4.0.2-64bit.msi， 首先出现如图 2.7 所示的安装准备界面，稍后会出现如图 2.8所说的安装初始界面。

# 创建Git版本库

通常情况下，我们需要设置 master 分支被保护，这可以使得 master 分支的修改在可控制范围之内。 按图 3.14 所示， 进入到图 3.15 所示的界面中设置分支保护，请设置 master分支保护，如同图 3.15 那样。这可能被默认设置，当然，你也可以设置其它分支被保护，被保护的分支除了项目管理员之外，不能直接被合并或推送，其他开发者想要修改被保护的分支需要通过发起合并请求来完成。

# Git及Gitlab设置

## Git设置

要开始使用Git，需要进行一些配置，主要的配置涉及到自己的用户名，邮箱，和对文本中的回车换行的处理。因为是在 windows 下工作，且多人协作，版本库又可能是由 svn版本库迁移而来，所以应该禁用 Git 的自动回车换行转换。 配置方式有命令行和 TortoiseGit两种，本文主要讲解TortoiseGit方式。

## Gitlab设置

### 设置在 GitLab 上的 SSH 密钥

配置 SSH 密钥可以让Git直接使用SSH和 GitLa上的Git版本库服务器通信，进行提交，获取，拉取等操作。 要配置自己在 GitLab上的SSH密钥，首先需要生成自己的密钥。

#### SSH 密钥简介

SSH 是目前较可靠，转为远程登录会话和其它网络服务提供安全性的协议。 GitLab 支持通过 SSH 访问其上的 Git 服务器，从而提供更快的传输速度和更好的安全性。 GitLab 上的 SSH 服务仅支持通过密钥进行认证，密钥有私钥和公钥组成，其中私钥里面包含了公钥的内容，公钥是公开的，保存在服务器，私钥用户保存用于认证。公钥和私钥统称为密钥对，其中公钥和私钥是一一对应的，但是从公钥推导出私钥是很难完成的。所以可以吧公钥放在服务器，用户使用私钥认证。

#### 生成自己的 SSH 密钥

和 Linux 一样， ssh-keygen 命令提供了生成密钥对的方法，首先启动 git-bash.exe，按照程序清单 2.2 所示执行 ssh-keygen 即可在用户目录下的.ssh 文件夹中生成公钥和私钥。

注意执行过程出现y/n的选项时，请输入y以便继续下去，同时出现Enter same passphrase时请按键盘上空格键以执行流程继续。

执行成功，此命令生成的私钥文件名为 id\_rsa.，公钥文件名为 id\_rsa.pub。密钥十分重钥，默认生成在用户目录，生成后请自行备份，以免因重装系统造成密钥丢失，造成需要重新生成ssh密钥才可以访问Gitlab。

#### 将公钥上传至 gitlab 服务器

1. 注册并登录Gitlab服务器

打开 Chrome 等非 IE 浏览器，输入网址 http://192.168.1.188/开始访问 Gitlab，首先出现的是注册界面，如图 2.30 所示，按提示进行注册后输入用户名和密码后登陆。登陆后的界面如图 2.31 所示。

1. 设置ssh密钥

请打开用户目录下.ssh 文件夹，用记事本打开其中 id\_rsa.pub 文件，并复制其内容，然后在浏览器进行如图 2.32 所示操作， 以便打开个人设置界面。

接下来进入个人设置主界面如图 2.33。 请单击图中鼠标位置的“SSH Keys”进入 SSH密钥设置界面，如图所示。

在点击出现后的界面当中，SSH 密钥设置界面的 key 区域粘贴前面复制的公钥，然后设置你想要的标题（title），然后单击鼠标位置的“Add key”按钮， 接下来会出现图 2.35 所示的 key 添加完成界面。再次单击 SSH Keys 可以回到密钥设置主界面，在下方以及可以看到添加成功的密钥了，如图 2.36 所示

1. 将 **SSH** 私钥转化为 **ppk** 格式

使用 TortoiseGit这个图形化工具代替Git命令行界面进行操作时，需要将前面生成的私钥转化为 ppk 格式。因为因为TortoiseGit使用了 TortoisePlink.exe来管理 SSH，而 TortoisePlink.exe只识别 ppk格式的私钥。

这个转化具体来说要用到 puttygen.exe， 3这个工具位于 TortoiseGit 的安装文件夹下 bin文件夹中。启动 puttygen.exe，其界面如图 2.41 所示

接下来，请按图 2.42 所示，选择菜单 Conversions->Import key， 在弹出的“加载私钥”对话框中，选择前面生成的私钥文件，如图 2.43 所示

加载私钥成功后，可以看到私钥加载成功后的界面，如图 2.44 所示，可以看到 key 区域已经有了内容。鼠标单击图 2.44 所示的“保存私钥”按钮，会接下来弹出如图 2.45 所示的保存私钥对话框，选择设置你想要的保存路径和文件名，这里依然保存在用户目录下.ssh文件夹下，文件名为 gitlab.ppk。 这样 SSH 私钥就成功转换为 ppk 格式

#### 克隆一个版本库证明 ssh 密钥设置成功

1. 获取版本库的地址

在图 2.36 添加 ssh 密钥后又重回 ssh 密钥设置界面中单击所图 2.36 示图标，就可以回到 gitlab 的主界面，如图 2.38 所示。 请按图 2.38 鼠标指示的位置单击一个自己有权限的版本库，进入到版本库的主界面。

在图 2.36 添加 ssh 密钥后又重回 ssh 密钥设置界面中单击所图 2.36 示图标，就可以回到gitlab 的主界面，如图 2.38 所示。 请按图 2.38 鼠标指示的位置单击一个自己有权限的版本库，进入到版本库的主界面。

1. 克隆版本库

可以命令行或使用TortoiseGit的方式来克隆版本库，使用TortoiseGit的方式克隆版本库操作更加友好，建议初学者使用。

首先从桌面计算机，进入到你想要克隆版本库用于工作的目录，在空白处单击右键， 选择 Git 克隆，如图 2.46 所示。

接下来会弹出所图 2.47 示的克隆对话框。 请按图中红色文件的标注，正确填写版本库的 URL，克隆下来版本库在本地的目录，和私钥的位置，然后单击确定按钮。

接下来会弹出一个 Git 命令进度对话框，如图 2.48 所示，并且很快，如果以前没有使用 TortoiseGit+SSH 密钥的形式访问过 Gitlab 的 Git 版本库，会弹出一个如图 2.49 所示的TortoiseGitPlink 安全警告，其大意是服务器的密钥没有在注册表中缓存过，可能没办法确认服务器就是信任的主机，如果你信任该主机，请单击是将主机密钥加入 putty 缓存，这里我们选择“是”。

稍待， Git 命令进度对话框会显示克隆成功，如图 2.50 所示。

# 和GitLab协同作战

和 GitLab 协同工作的主要分为两大类型，第一类是拥有主分支写权限（项目管理员），第二类是开发者，不能直接 push 到master主分支,因为master主分支一般是受保护。

## 版本库管理员的工作

### 版本库的初始化工作

在一个新项目开始的时候， 其版本库为空， 项目管理员需要进行初始化工作，这些工作通常包含项目资料，工具和最基础源码的提交。这样版本库中才会存在 master 分支。 这些操作和通常的 Git 提交推送过程没什么两样。

#### 使用TortoiseGit 命令行进行提交推送添加

TortoiseGit 提交推送需要添加，提交，推送三个步骤，如下阐述所示。

1. 添加

具体来说，如图 3.1，首先在要添加的项目上单击右键，依次选择 TortoiseGit->添加(A)，接下来会出现添加对话框， 如图 3.2 所示，请选中要添加文件的复选框，单击确定，稍等，就会出现如图 3.3 所示的添加完成的对话框。单击确定即可

1. 提交

添加完成后，就可以提交，在工作根目录的空白处单击右键，选择 Git 提交，如图 3.4所示。注意菜单 Git提交后面跟着的字符串表明了你当前的分支，也就是你所要提交的分支。

接下来出现的是提交对话框，如图 3.5 所示。 填写日志，确认要提交的文件后，单击提交，稍等，就可以出现如图 3.6 所示的提交完成对话框了，在此对话框上单击关闭按钮。

1. 推送

同样在在工作根目录的空白处单击右键，选择 TortoiseGit->推送(H)， 如图 3.7 所示。

接下来会弹出推送对话框，如图 3.8 所示。 确定远端信息，本地和远端的分支之后单击确定， 稍待，即可出现推送完成对话框。

## 开发者和 GitLab 协同工作

### 克隆项目版本库

始接触这个项目，首先要克隆这个项目的版本库，在 git-bash 中切换到工作目录，然后执行如程序清单 3.3 所示的命令，即可克隆项目版本库， 程序清单 3.3 不仅展示了克隆命令，还展示了克隆过程。

### 开发者典型的工作流程

由于 GitLab对 master分支进行了保护，对 master分支的修改通常由项目管理员来完成，所以对开发者来说，每次修改都需要基于 master 分支进行修改， 又不能直接修改 master，所以通常来说都是基于 master 新建一个工作分支进行修改，然后推送这个新分支，然后在GitLab 上基于新分支发送合并请求，具体的操作流程依次如下所述。

#### 从远端服务器更新

要基于远端服务器的 master 进行修改，首先需要从远端服务器更新，以便获取最新的修改，然后再对本地 master 分支应用服务器上的更新， 利用 git fetch 和 git rebase 可以实现这样的更新，如程序清单 3.4 所示。

#### 创建并切换到新的工作分支

接下来需要创建并切换到新的工作分支，以便开始新的编码工作， 使用一条 git checkout命令就可以实现这个功能，如程序清单 3.6 所示。

#### 编写代码

按照自己的工作要求编写相关代码并测试，此处不详述。编码自测通过后，请添加提交到自己的本地工作分支上。

#### 再次更新服务器上的内容

之所以要进行此步骤，是因为在编码测试期间，服务器上的内容可能又发生了更新，你可能需要下载更新内容，并与自己的修改合并，看看自己的修改在服务器的新内容上能否生效。 此次更新需要发生在工作分支上，因为自己也进行了修改，所以可能会发生冲突，所以操作方式与前次也有所不同

首先，请获取服务器上的修改，用的命令依然是 git fetch，如程序清单 3.7 所示。

然后需要将本地的修改和服务器的修改合并，为了减少网状结构和方面以后处理合并请求，这里应该使用 rebase 而不是 merge 来进行操作，为了方便处理冲突，这里推荐使用TortoiseGit 来操作，操作应该在本地版本库工作根目录下进行。 首先请按图 3.16 所示操作进入 TortoiseGit 的 Rebase 界面，如图 3.17 所示。

在图 3.17 中，请首先确保上游正确，然后单击开始变基按钮，变基过程便会自动进行，如果遇到冲突的文件，就会如同图 3.18 那样的提示。可以双击鼠标位置，在弹出的窗口中解决冲突， 但是也有一种更加方法点的做法是再鼠标位置单击右键，选择使用现有解决有冲突的文件， 如图 3.19 所示。

操作后文件的状态就会变为已修改，然后在图 3.18 鼠标位置双击，就会出现远端修改和你的修改的对比，如图 3.20 所示。其中做成为远端的文件，右侧为你修改的文件，你可以对比变化，将右侧的文件修改为能够体现双方修改，并能正常工作的文件。

图 3.20 中，左边表示远端有人添加了“Hello World2”的打印，右边表示你添加了“Hello World3”的打印，经过思考，将右边的内容改成了两个语句都存在即两句话都打印。 单击保存后可以退出，注意左边的文件是不能修改的。解决完所有冲突后在 Rebase 对话框上单击提交按钮， 稍后机会出现成功的提示，如图3.21 所示。

单击完成按钮后可以退出变基对话框， 然后对于新的版本请编译测试没问题之后，就可以将这个工作分支推送到远端了。

#### 推送工作分支到远端

用如程序清单 3.8 推送到远端所示的命令即可。

#### 发起合并请求

推送工作分支到远端之后，可以在 GitLab 上发起合并请求了。可以单击图 3.22 所示图档进入发送合并请求的界面，如图 3.23。 在图 3.23 的鼠标位置单击，即可进入新的合并请求界面。

在图 3.24 所示的新的合并请求的初始界面中，左边鼠标处是合并的源分支，右边是目标分支，请确认信息正确后，单击 Compare branches and continue。  
接下来会出现合并请求的信息页面，如图 3.25。请填写好描述之后， 在页面最下方单击如图 3.26 所示位置的提交按钮

接下来会出现如图 3.27 所示的，提交完成页面。

合并请求完成后，请和项目管理员沟通，让管理员合并到主分支。如果请求被驳回，请修改后再次从第 5 步开始重新操作。

## 项目管理员的工作之处理合并请求

项目管理员的重要工作，即是处理开发者发起的合并请求。 首先登录 GitLab，然后进  
入到项目版本库的主页，以及点击“Merge Requests” ->Open，就会出现当前需要处理的合并请求。如图 3.28 所示，鼠标位置就是最近的合并请求，单击进入合并请求详细页面。

如图 3.29 所示， 在确认提交，更改， pipelines 的运行结果没有问题后，在确保删除源分支选项被勾上之后，就可以单击鼠标位置的按钮接受合并请求了。

# 项目优劣

## 优势

如果基于简单易用的STM32L系列的芯片平台，推出板载 GPS 模块，温湿度传感器，环境光传感器，烟雾传感器等诸多实用硬件的NBIoT开发板及相应的资料及教程，有利于帮助客户先期熟悉NB网络，然后快速设计出基于NB网络的产品，抢先占领市场。我们公司而因此在NBloT当中积累粉丝，提高品牌知名度。同时现阶段积累下来的NBloT技术，可能以后的产品项目当中能使用到。

我们比较直接的竞争对手安富莱及野火并没有推出NBloT相关的资料，如果我们推出来，这在竞争当中会取得一些优势。

## 劣势

不是最早推出NBloT相关资料的公司，可能会失去了一些先机。同时需要投入一定的人力跟进研究。

# 项目实施

根据目前的调研情况，我们可以参考谷雨的开发板及资料，在项目前期，作出以下的项目实施计划，如如表 6‑1所示。

表 6‑1 项目实施计划表

|  |  |
| --- | --- |
| 日期 | 安排内容 |
| 6月18日-7月2日 | 画硬件原理图,模组外围电路，MCU主控电路，传感器电路，GPS模块电路，调试相关电路 |
| 7月2日-7月15日 | 利用现有板子及模块写完例程 |
| 7月16日-7月31日 | 接入电信云的基础例程 |
| 8月1日-8月15日 | 接入电信云的高级例程(项目实战)，如温湿度检测 |
| 8月16日-8月30日 | 教程编写及资料整理,同时录制教程视频 |

NB-EK-L476是谷雨物联网推出的NB-IOT开发板，基于简单易用的STM32L476芯片平台，

板载 GPS 模块，温湿度传感器，环境光传感器等诸多实用的硬件外设，可帮助客户快速设

计出基于 NB 网络的产品，抢先占领市场。尤其现阶段， NB-IOT 是个日新月异的领域，所以

分秒必争！ 表1.1特性参

后期，公司的原子云如果支持CoAP协议的话，可以出NBloT接入原子云的教程。如果能获得华为审批的通过，也可以出接入华为云的教程。

注：以上安排仅供参考，可以根据实际情况进行调整。

# 项目实施难点

## 政策难点

需要向运营商的物联网政企部门申请SIM卡，即NB卡。NB卡根据运营商现有的政策需要绑定接入云平台的IP地址。结合公司的情况，我们有可能需要申请绑定接入（原子云，电信云，移动云，华为云）服务器的ip地址。同样的，也需要向运营商的物联网政企部门申请云对接平台账号。如果需要接入华为云，还需要向华为申请华为云对接账号。这些流程审批严格、繁琐并且比较耗时，最后还有可能还不被批准。

同时，现在NBloT覆盖率还不是很全，电信虽然说在全国已经实现商用，但偏的地方没有信号，绐调试带来难度。移动还没有实现全面商用，还在测试阶段。

## 软件难点

根据目前的了解，软件上存在以下一些难点：

1. NBlot模组AT指令流程的封装。
2. 电信云接入平台调试，包括一个插件的调试。

## 硬件难点

暂时未知，根据开发的情况而定。

# NB卡与云平台账号申请

## 对接云平台账号申请

客户可自行登录“天翼物联产业联盟”微信公众号填写开放平台测试账号申请，如果不懂申请，可以向电信政企部门对接的客户经理联系如何申请。如果客户自行申请，一般流程如下所示：



图 8-1查找微信公众号并进入



图 8-2点击联盟服务中平台对接认证



图 8-3完成表格填写



图 8-4完成表格填写

注意，NB测试卡一般是以10649开头的13位号码，如果客户掌握NB卡的服务密码，可以通过提供NB卡背面的智能卡号查到，否则只能通过联系当地或对接的客户经理来获取。如果客户通过微信公众号线上申请时，不知道当地客户经理或者NB卡对接的客户经理，可以忽略客户经理相关项的填写，申请也有可能通过。如果失败，请联系当地的客户经理或对接的客户经理处理。

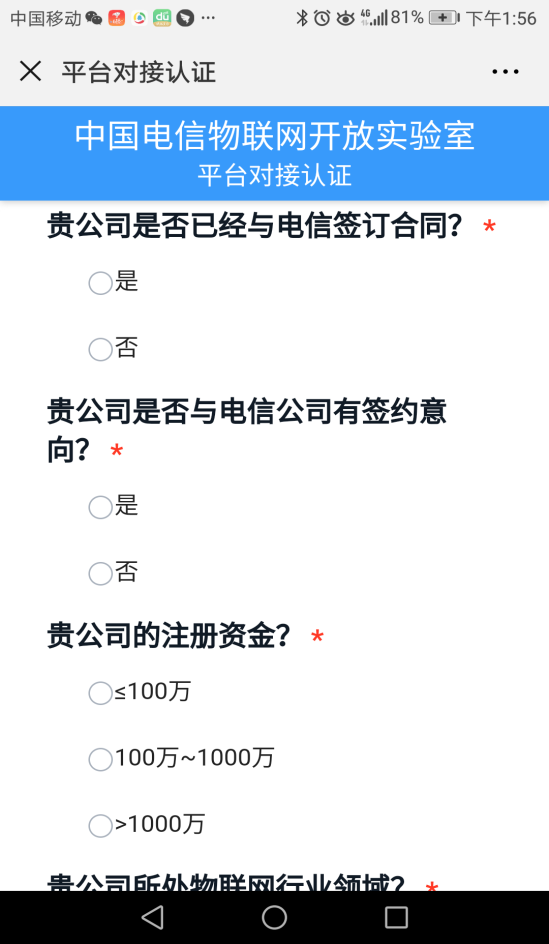


图 8-5完成表格填写

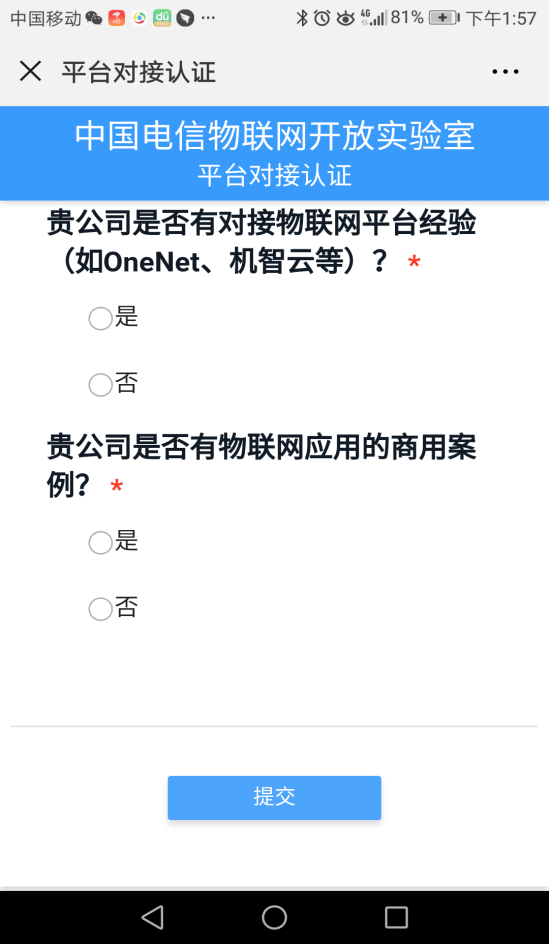


图 8-6完成表格填写并提交

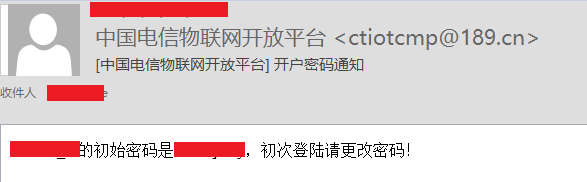


图 8-7申请通过后，收到账号和初始密码

## NB卡申请

NB卡申请，需要到当地电信政企部门找客户经理申请。如果不知道当地电信政企部门在哪里，建议到附近较大的电信营业厅资询了解。如果去规模较小的电信营业厅，可能不了解这个。

目前申请NB卡，只能以企业单位的名义，不支持个人申请，同时目前只能线下申请。企业单位相关人员到当地电信政企部门办理NB卡申请时，需准备相应的资料，如企业单位的营业执照，公司性质证明等相关材料。

至于申请到的NB卡，需要为NB卡绑定固定的IP地址。一般来说，电信的NB卡默认只绑定电信云平台的服务器IP。如果想要绑定除电信云平台的服务器IP之外，需要与电信政企部门客户经理讨论，看他是否给你办理，或者说他有足够的权限给你办理这项业务。同时是否能为申请到的每一个NB卡同时开通一个对接的云平台账号，也要与电信政企部门客户经理讨论，看他是否给你办理，或者说他有足够的权限给你办理这项业务。

## 服务热线

一些流程详细了解，可以打***4008285656***电信物联这个官方客服电话进行了解。

# 联系我们

1. **官方店铺**

官方店铺1：<http://shop62103354.taobao.com>

官方店铺2：<http://shop62057469.taobao.com>

1. **资料下载**

资料链接：<http://www.openedv.com/thread-272133-1-1.html>

**3、技术支持**

技术论坛：[www.openedv.com](http://www.openedv.com)

官方网站：[www.alientek.com](http://www.alientek.com)

联系电话：020-38271790