MN316 R14版OpenCPU FOTA测试经验文档

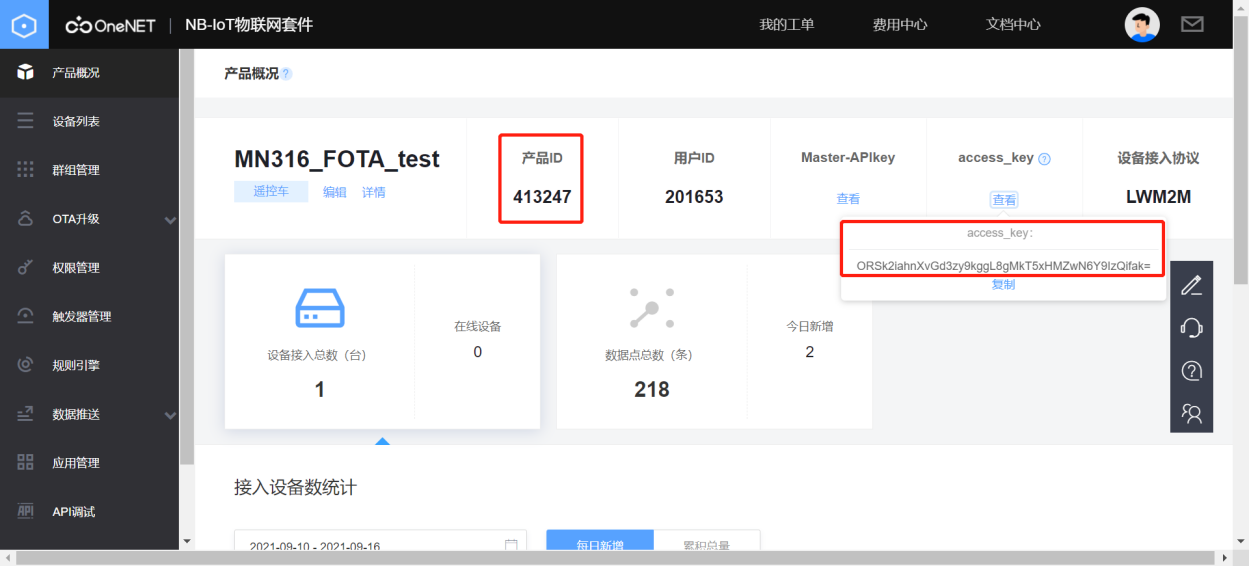
——智能模组部FAE 何玮

2021年9月22日

前面讲解了MN316 R14版本OpenCPU的开发环境搭建，这次对其FOTA功能进行测试，目前支持移动OneNET平台和电信CTWing平台的FOTA升级，都是通过差分包方式升级。不同的是，OneNET是模组侧触发升级，CTWing是平台侧触发升级。

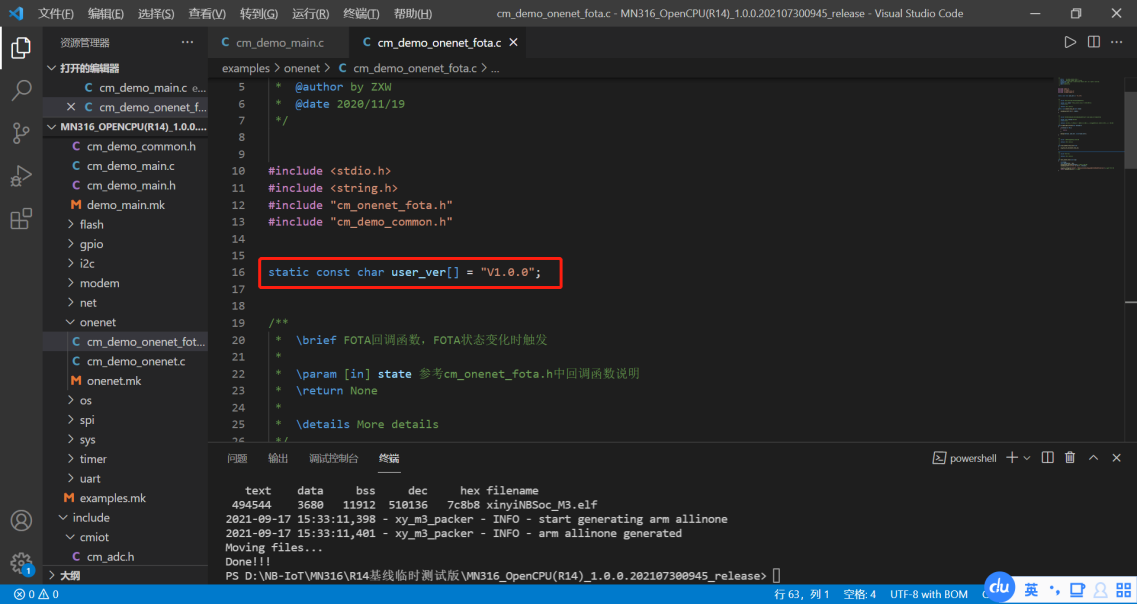
**一、通过移动OneNET平台进行FOTA**

1、登陆平台后，在NB-IoT物联网套件里创建产品，记录下产品ID和access\_key，再将测试用的MN316模组添加到该产品下



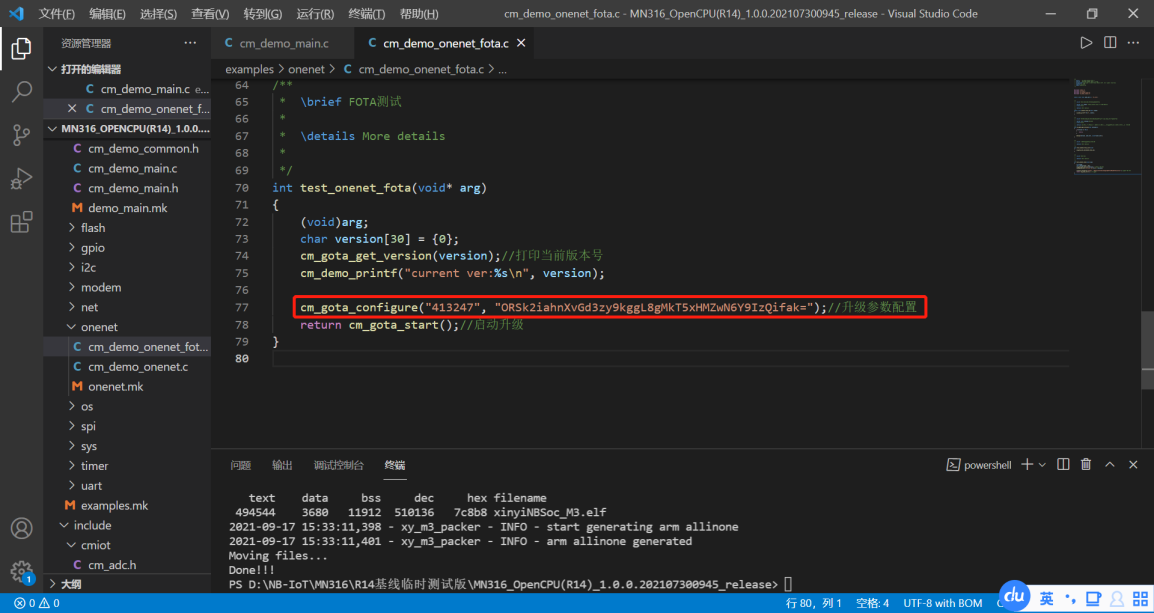


2、用VS code打开MN316 R14版的SDK（测试版本：MN316\_OpenCPU(R14)\_1.0.0.202107300945\_release），打开cm\_demo\_onenet\_fota.c文件，在static const char user\_ver[] = "V1.0.0"修改用户SKD版本号，demo默认设置的是V1.0.0



此版本号由用户自定义，后续在OneNET平台上创建的软件版本号必须与此一致。

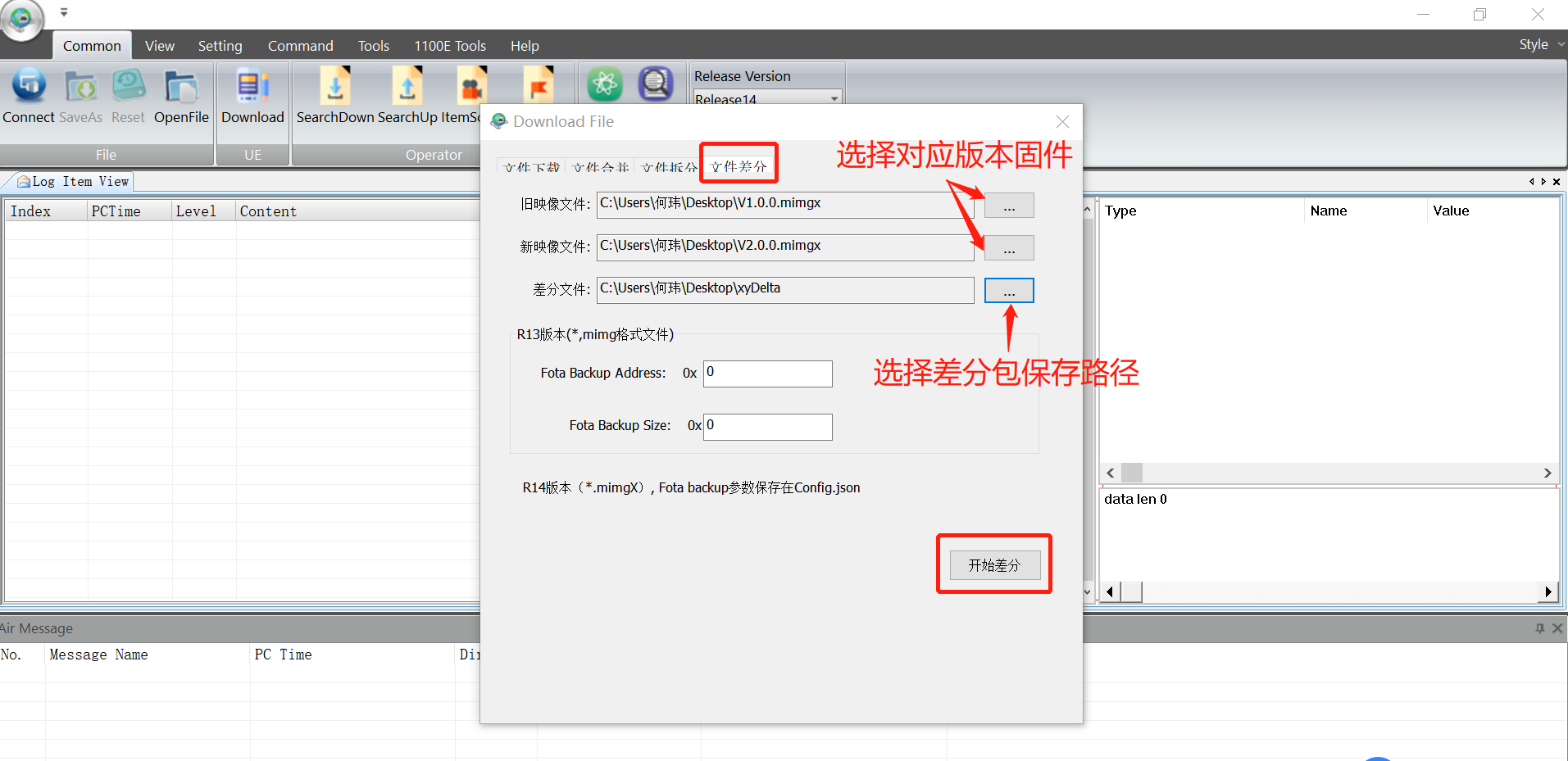
在该文件最后是demo测试程序test\_onenet\_fota()，包含打印当前SDK版本号、升级参数配置、启动升级共三个步骤，需要在升级参数配置接口cm\_gota\_configure()里添加刚刚创建的产品ID和access\_key



我们将该版本SDK当做初始版本，编译SDK，合并成.mimgx格式文件，为了便于区分，可以将烧录文件名改为V1.0.0.mimgx。

在更新了程序后，修改SDK版本号，例如改成V2.0.0，将此版本当做目标版本，再编译SDK，将烧录文件名改成V2.0.0.mimgx。

3、使用研发最新Logview工具（版本：Logview\_1.3.210501.0\_CMVer1.2.1）进行差分包制作（若使用以前版本工具，制作的差分包将升级失败）：



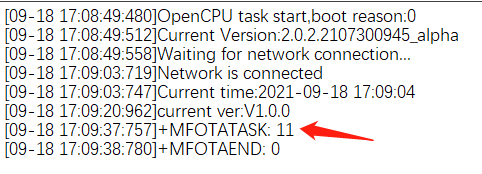
生成两个差分包文件：xyDelta和xyDelta\_downgrade

xyDelta用于从V1.0.0升到V2.0.0，xyDelta\_downgrade用于从V2.0.0降到V1.0.0

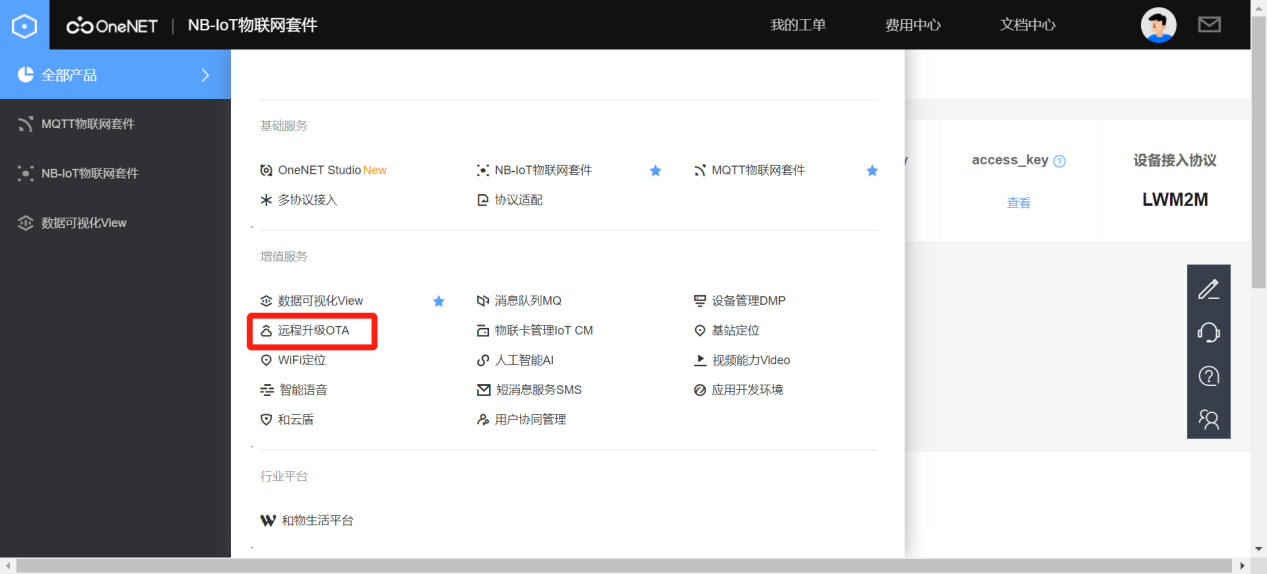
但是，由于OneNET平台能识别的文件类型只有.zip/.rar/.bin，故我们可以将差分包文件改为.bin格式：

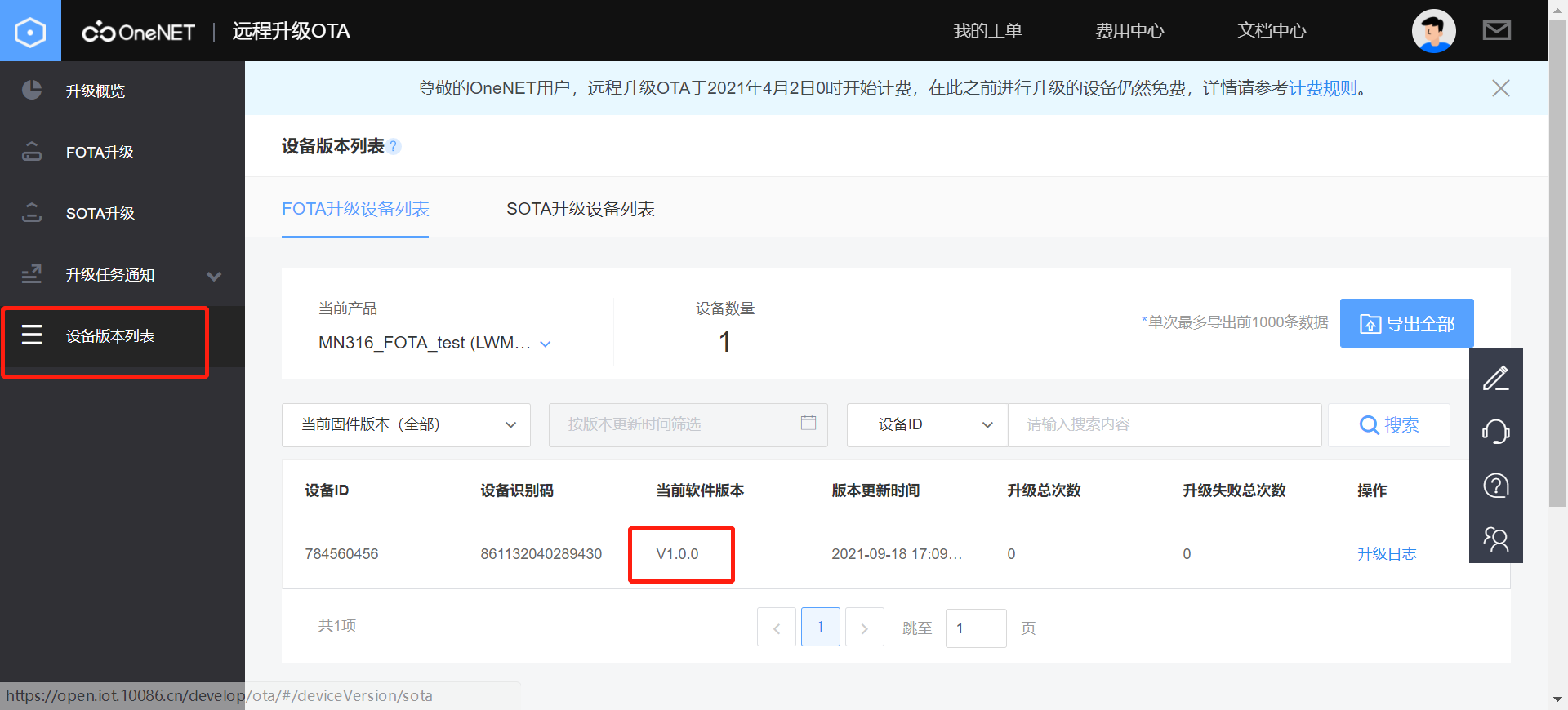


4、将初始版本V1.0.0.mimgx烧入模组，MN316 OC版本第一次在OneNET上FOTA升级时，需要先触发一次FOTA升级，即模组与平台进行一次交互，才能将用户SDK版本号同步到平台上被平台识别，否则FOTA时无法找到设备。首先开发板给模组上电，待驻网成功打印后，串口发送“GOTA”,模组将执行cm\_gota\_start()来触发FOTA升级，模组会打印与平台交互信息：



这里模组上报+MFOTATASK:11的意思是平台上没有对应的升级任务，表示模组已经和平台交互过了，此时在平台上就可以查询到SDK版本号：





5、添加升级包，分别创建升级前后两个版本的升级包，一定要确保写入的目标版本号与步骤2中设置的SDK版本号一致







然后根据对应的目标版本选择添加相应的差分包:在目标版本V2.0.0下添加升级差分包文件xyDelta.bin，在目标版本V1.0.0下添加降级差分包文件xyDelta\_downgrade.bin

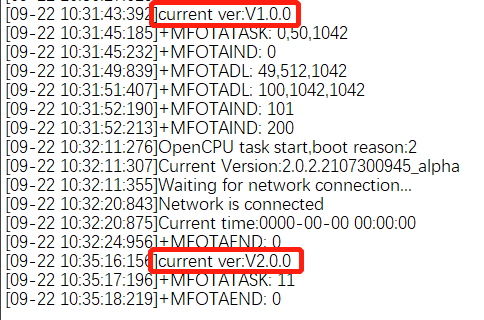


点击目标版本V2.0.0后面的“升级设备列表”，选择初始版本和设备然后点击“添加”，此时平台测操作完成：

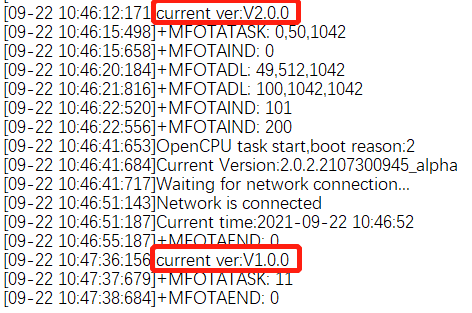




6、再次给模组串口发送”GOTA”，模组打印当前SDK版本号并启动FOTA流程，待升级完成后模组自动重启，再给模组串口发送”GOTA”，模组打印升级后的版本号，由此可确认模组FOTA成功：

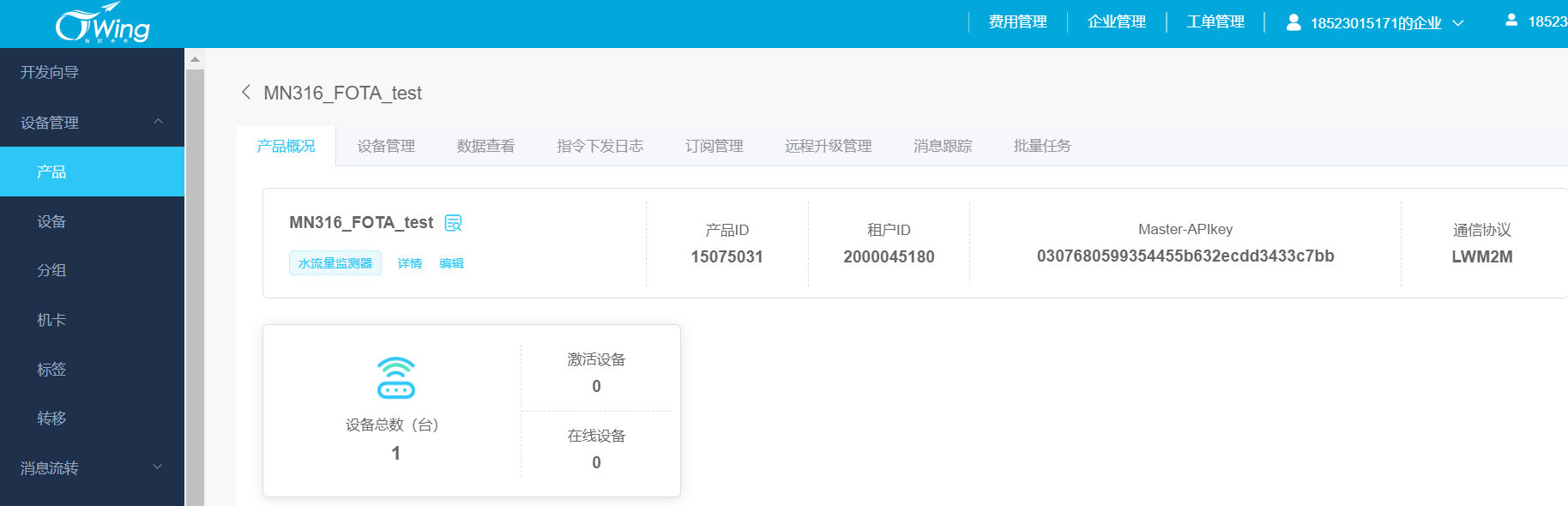


同样的操作方式，在目标版本V1.0.0下，模组也可以从V2.0.0降级到V1.0.0：



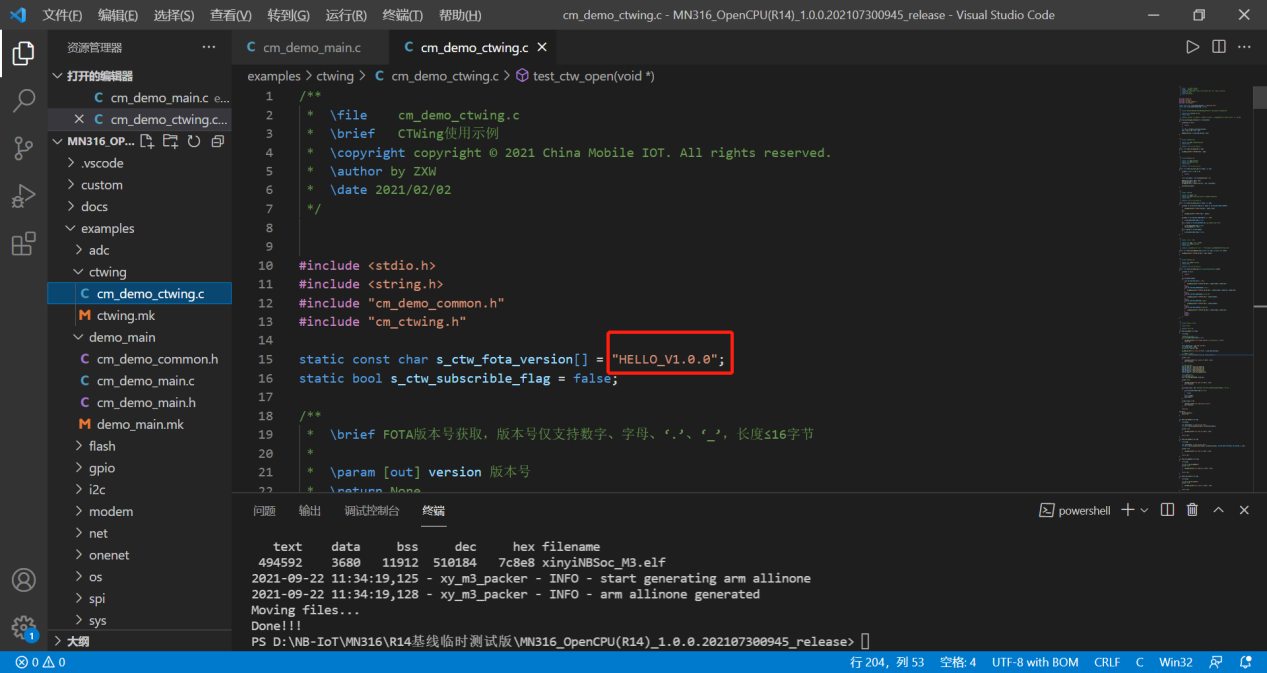
**二、通过电信CTWing平台进行FOTA**

1、登陆平台后，创建一个LWM2M协议的产品并将测试用的MN316模组添加到该产品下：





2、用VS code打开MN316 R14版的SDK（测试版本：MN316\_OpenCPU(R14)\_1.0.0.202107300945\_release），打开cm\_demo\_ctwing.c文件，在static const char s\_ctw\_fota\_version[] = "HELLO\_V1.0.0"修改用户SKD版本号，demo默认设置的是HELLO\_V1.0.0



（需要说明的是，CTWing平台的FOTA并不需要识别SDK版本号，此处设置的版本号仅用于串口打印，方便用户观察是否升级成功）

我们将升级后的SDK版本号改为WORLD\_V2.0.0，分别编译两个版本的SDK并生成对应差分包（与前面OneNET部分的操作一样，此处不再赘述，只是生成的差分包不用修改文件格式，CTWing没有限制文件格式）



3、选择“远程升级管理”，点击“固件管理”，再“创建固件”：







固件名称、版本可以任意定义，固件文件选择刚刚生成的差分包文件xyDelta，点击确定后，返回“远程升级管理”，点击“创建任务”，选择刚刚创建的固件版本：

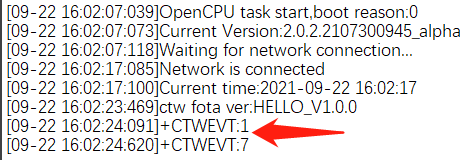






至此，平台升级任务创建完成

4、与OneNET平台不同的是，由于是CTWing主动发起FOTA，所以模组需要先登录注册到CTWing平台，模组需在平台上处于已激活状态。首先我们将初始版本HELLO\_V1.0.0固件烧录到模组并上电开机，待驻网成功打印后，串口发送“CTWOPEN”，模组上报如下图，则表示注册成功：



5、在“远程升级管理”界面，点击“升级设备管理”：



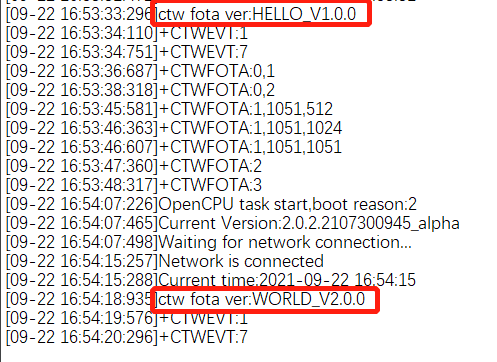
选中刚刚已激活的设备，点击“加入升级”，此时设备处于待升级状态：



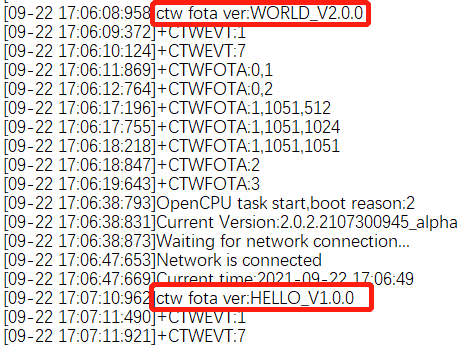
再返回在“远程升级管理”界面，点击“立即启动”：



6、模组重启或者注销CTWing后，串口输入“CTWOPEN”，打印当前SDK版本号并重新登录注册CTWing，开始进行FOTA流程，升级完成后模组自动重启，再次串口输入“CTWOPEN”打印升级后的SDK版本号，可以确认已升级成功



同样，上传固件xyDelta\_downgrade可以将程序从WORLD\_V2.0.0回退到HELLO\_V1.0.0



平台上也可以看到升级成功的提示

