**QGC地面站环境搭建及开发入门**

1. 环境搭建及编译

QGroundControl又叫QGC地面站，是一款跨平台的开源地面站，支持windows、linux、ios、安卓多平台。

本开发教程以windows系统（win7/win10）为开发平台进行环境搭建，编译生成windows版本（下称PC地面站）和安卓版本（下称安卓地面站）。

1. 准备内容

（1）开发电脑：win7/win10系统

（2）手机平台：安卓手机，运行内存>2G，已开启开发者权限和USB调试功能

（3）软件准备：

VS2017（下载及安装详见MP开发指南）

QT 5.12.6（下载地址：<http://download.qt.io/archive/qt/5.12/5.12.6/>）

Andriod Studio（下载地址：<https://dl.google.com/dl/android/studio/install/3.5.2.0/android-studio-ide-191.5977832-windows.exe>）

Jdk（下载地址：<https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>）

Ndk（下载地址：<https://dl.google.com/android/repository/android-ndk-r21b-windows-x86_64.zip>）

QGC源码下载（源码下载之前记得先下载git ，再新建一个文件夹，用来存放qgroundcontrol代码。在该文件下，点击鼠标右键，有一个Git Bash Here，然后输入如下命令：

// 源码下载：

git clone -b Stable\_V4.0 https://github.com/mavlink/qgroundcontrol.git --recursive

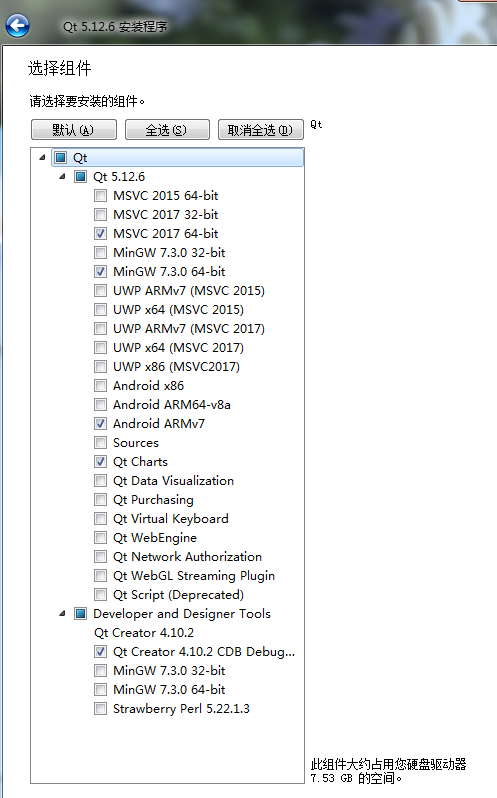
// 更新子目录

git submodule update）

以上软件均保存于网盘上，下载：XXX

1. PC版本地面站编译及错误处理
2. VS2017安装（详见MP开发指南）
3. QT安装

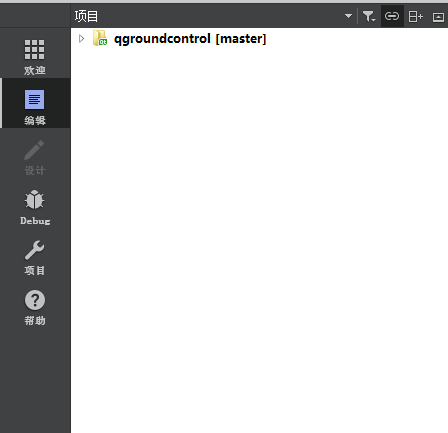
安装组件选择如下：



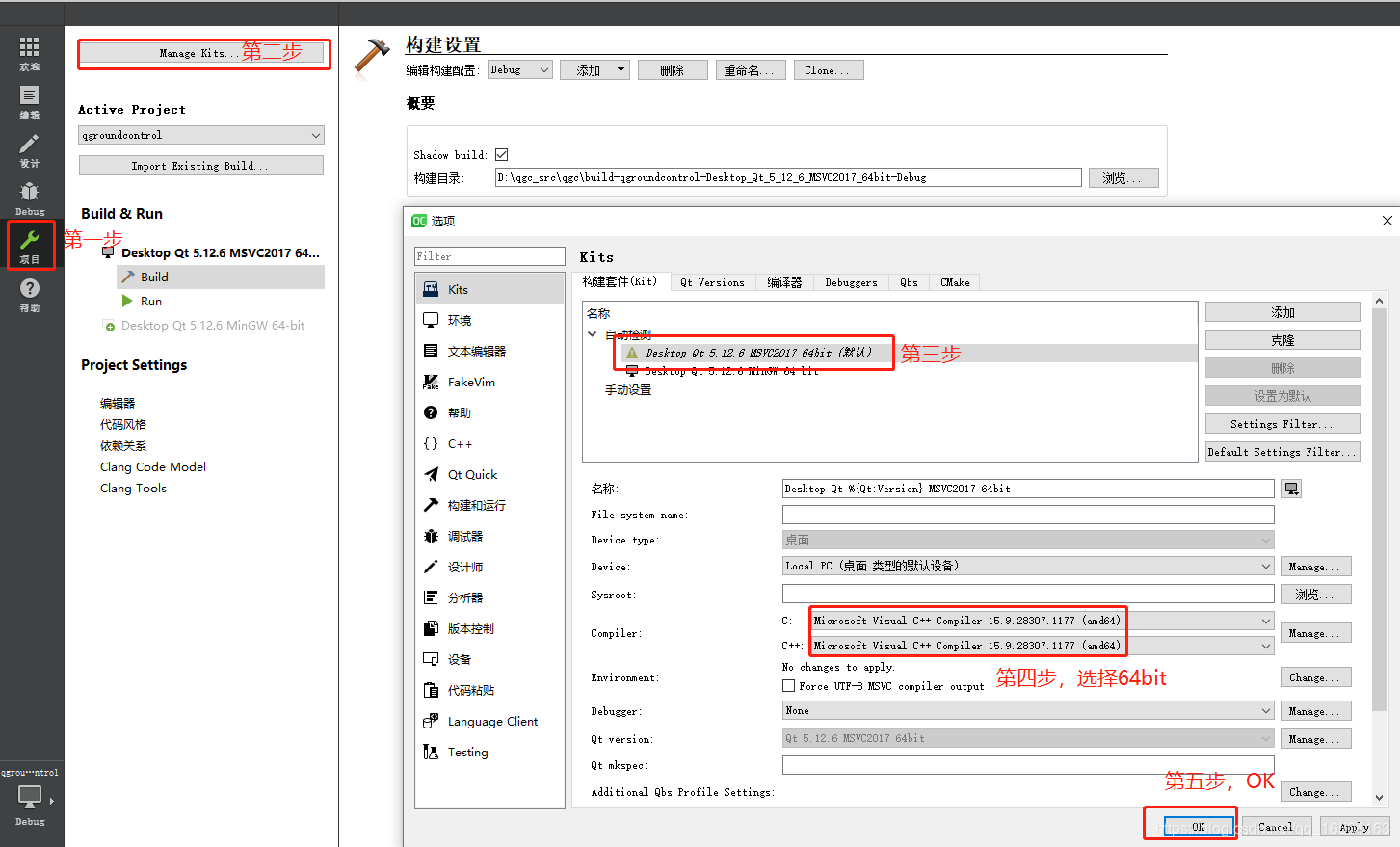
其他步骤按照提示进行即可

1. 编译源码

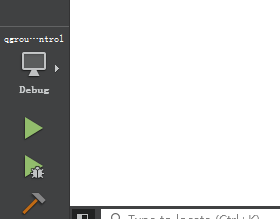
qt creator打开QGC工程（.pro后缀的文件）



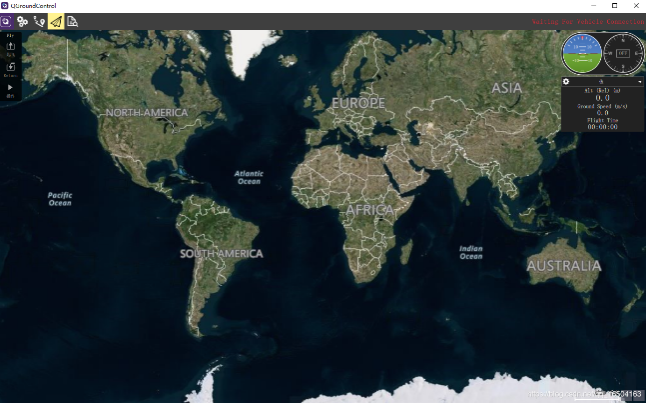
按照下面的配置，选择64位的，就是VS2017安装的编译器：



设置完后进入编辑，单击左下角三角符号运行。



运行完后结果：



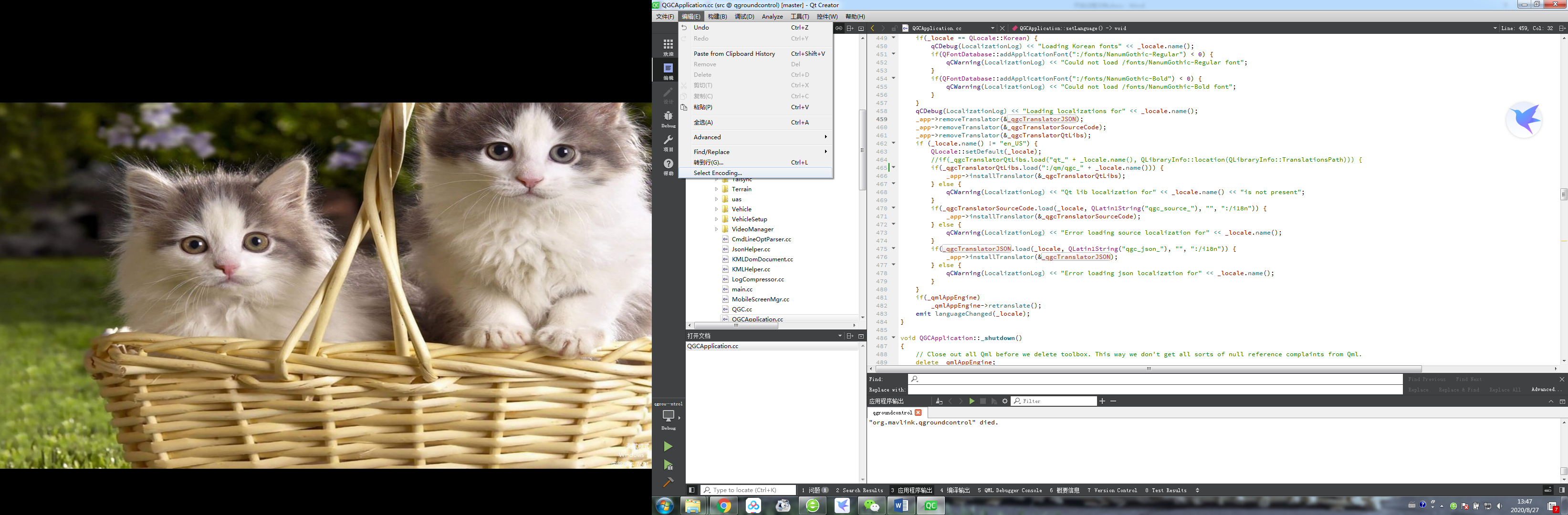
（4）编译报错处理

1. C2220：警告被视为错误，没有生成object文件

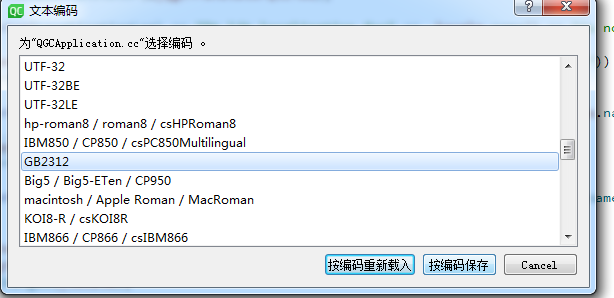


这个错误是编码格式引起的，需要改变文件编码格式，解决方法：

双击错误提示行，定位到报错文件，选择编辑—Select Encoding



选择GB2312，然后选择按编码保存，重新编译，其他文件再报错用同样方法处理。



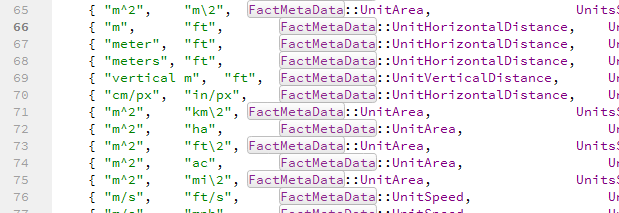
1. C2220：警告被视为错误，没有生成object文件（FactMetaData.cc文件，按上一步骤处理无效）

将文件中第65、71、73、75行中的u00B全删除即可

源文件：



改完后：



1. 调试无法进入断点，提示The selected debugger may be inappropriate for the inferior.

解决方法：下载windows sdk win10 sdk

只安装Debugging Tools for Windows

打开 工具-选项-Kits

安装sdk成功后我们可以看到 已经检测到了我们安装的cdb

打开工具-选项-构件套件

然后在qtcreator里面设置这个cdb,点OK重启qtcreator即可

1. 安卓版本地面站编译及错误处理
2. JDK安装

可以自定义安装路径，记住JDK安装路径，后面配置环境变量需要使用



设置并记住JRE安装路径（JRE即为Java运行环境）

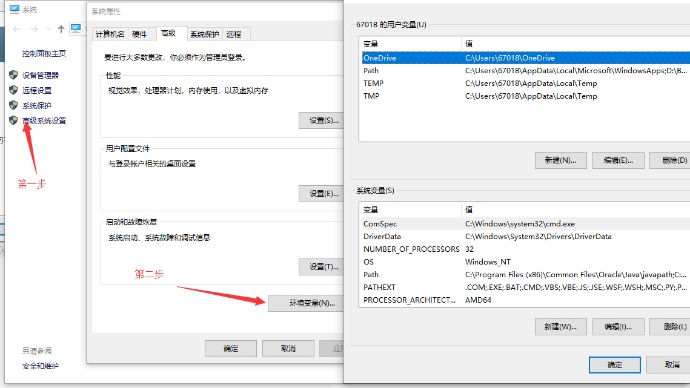


安装完成后点击关闭



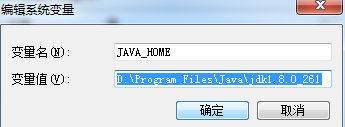
接下来进行环境变量的配置

**右键我的电脑，点击最下面的属性，之后再弹框里点击高级系统设置，再点击环境变量**



**新建->变量名"JAVA\_HOME"，变量值C:\Java\jdk1.8.0\_05（即JDK的安装路径）**

**编辑->变量名"Path"，在原变量值的最后面加上 %JAVA\_HOME%\bin; %JAVA\_HOME%\jre\bin**





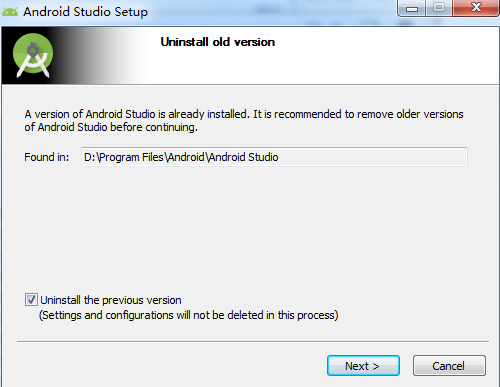
接下来进行验证，看是否成功安装了java

打开cmd，输入java -version观察；出现下面现象就表示OK

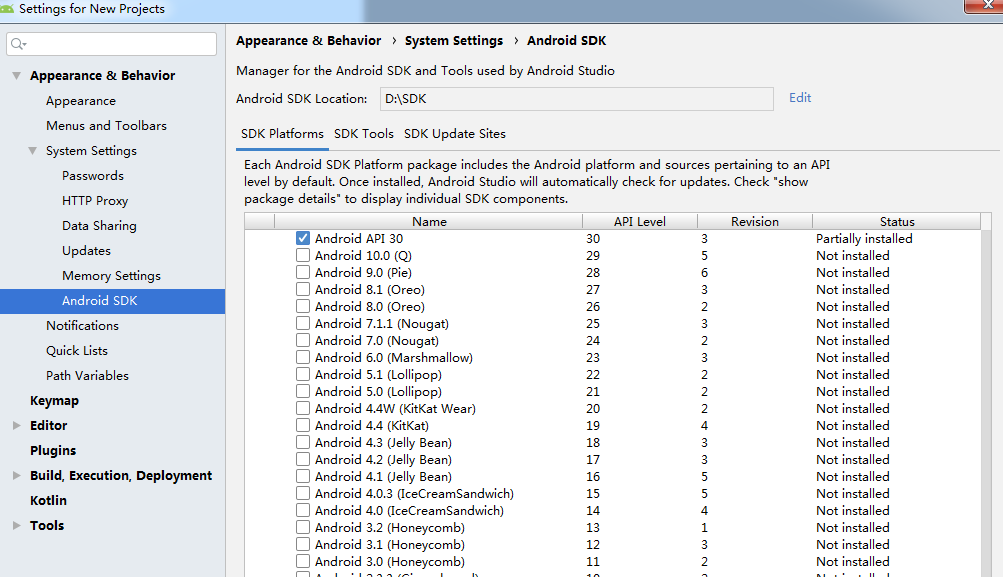


1. Andriod Studio安装

按照提示安装完成Andrio Studio



在android studio主界面搜索sdk manager，弹出如下界面，在其中可进行sdk的下载、升级、卸载等操作。更改安装位置，下载Android API 30。

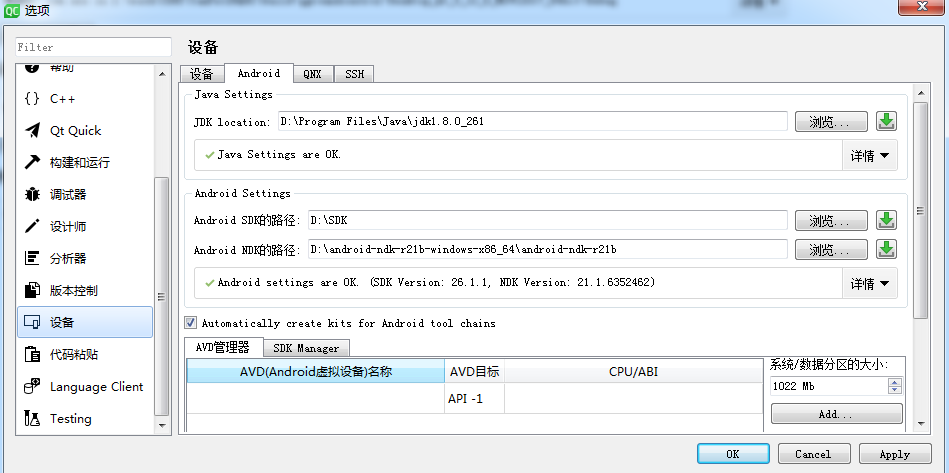


1. ndk的安装

NDK的安装很简单，将安装包解压即可，记住解压的位置。NDK的路径中不能有空格。

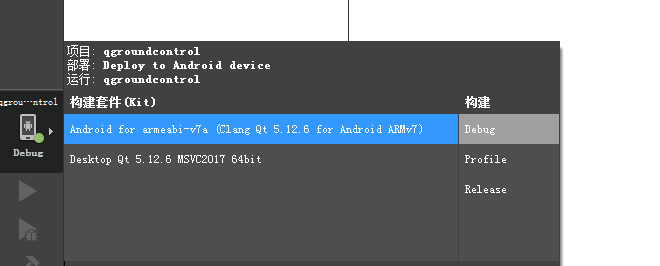
1. 设置qt环境

打开qt creator，工具—选项—设备—Android，在其中将JDK路径、SDK路径、NDK路径设置好。

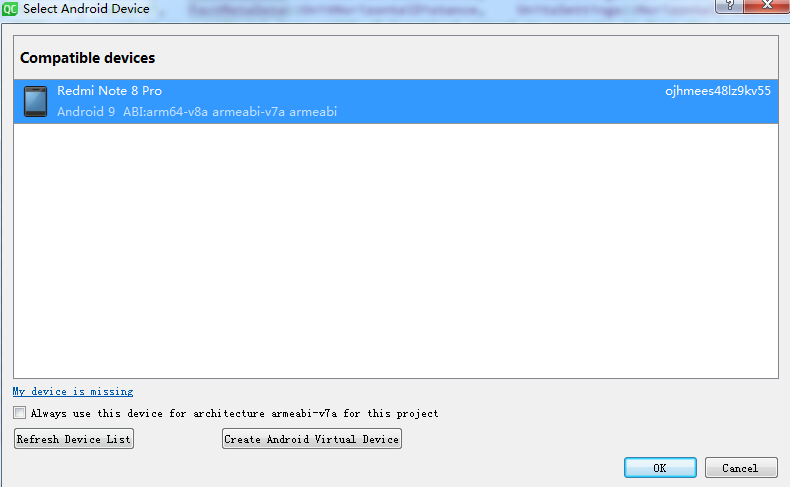


1. 编译源码

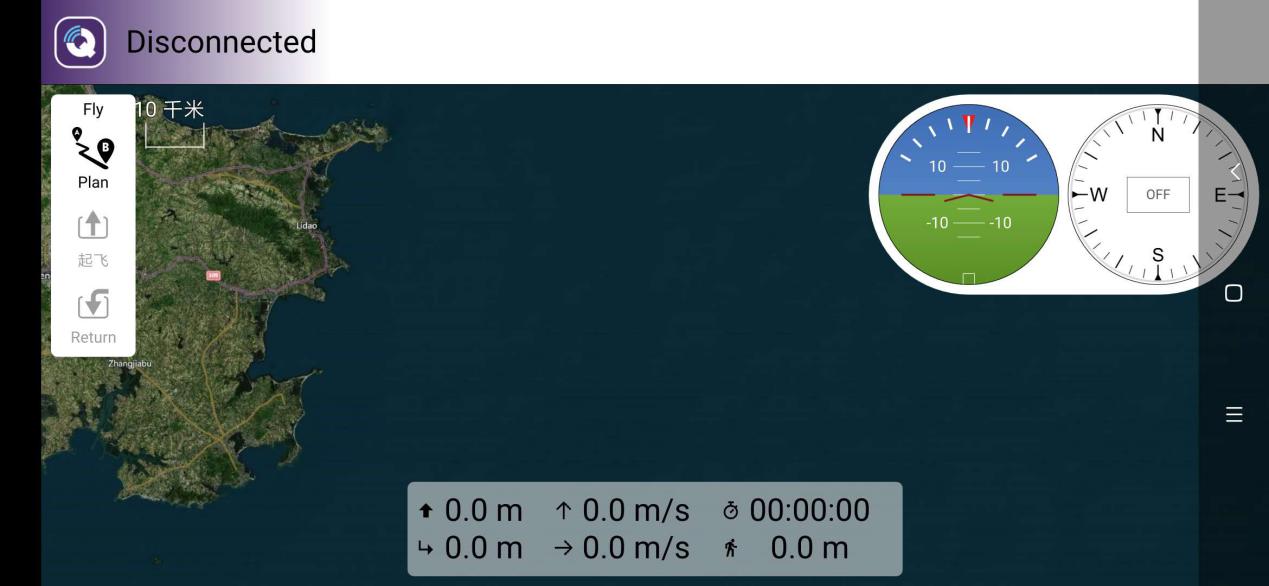
到编辑界面，Debug选项选到Android for armeabi-v7a



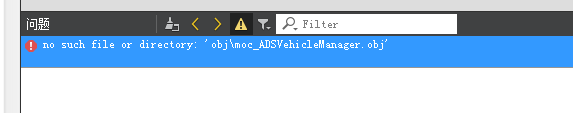
将手机使用usb连接到电脑，打开开发者模式和usb调试功能，点击运行，弹出窗口中选择使用手机。



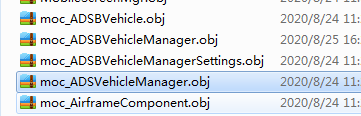
初次编译时间比较久，耐心等待软件安装到手机上。



1. 编译报错处理
2. No such file or directory:’obj\moc\_ADSVehicleManager.obj’



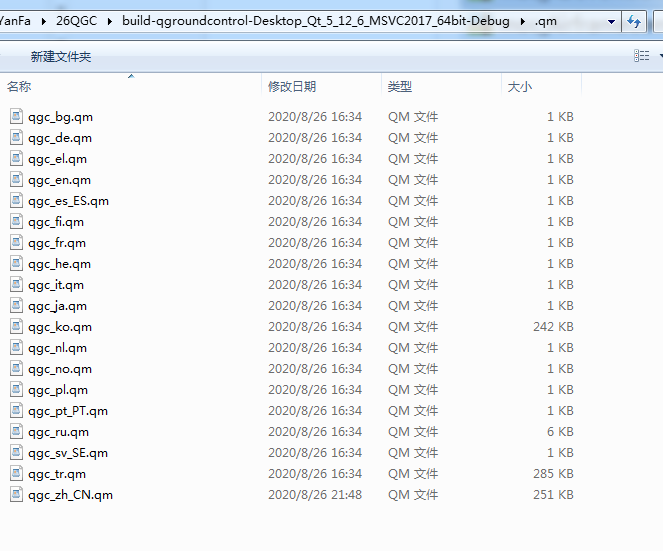
此报错原因可能还是因为编码格式的原因，导致寻找的文件与生成的obj对象名称不符，比如此条报错，实际生成的obj文件在编译文件里面，名称为moc\_ADSBVehicleManager.obj，复制重命名为moc\_ADSVehicleManager.obj即可，如下图所示。



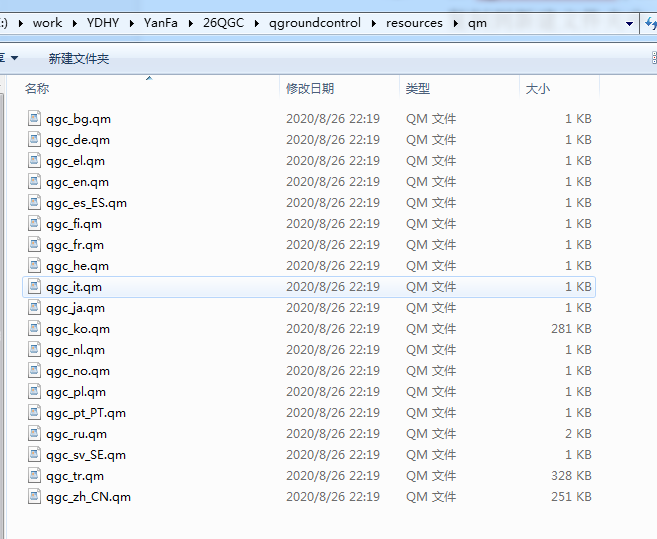
1. 地面站汉化

在QGC源码包中已经包含了汉化文件，但因为路径问题默认无法调用，导致无法选用中文包。

1. 在编译文件夹中找到.qm文件夹，里面的qgc\_zh\_CN.qm文件即为汉化文件



1. 在qgroundcontrol\resources文件夹中建立文件夹qm，将（1）中所有qm文件复制到新建文件夹中



1. 使用写字板打开qgroundcontrol\qgcresources.qrc

将如下字段复制到倒数第二行

<qresource prefix="/qm">

<file alias="qgc\_zh\_CN">resources/qm/qgc\_zh\_CN.qm</file>

<file alias="qgc\_bg">resources/qm/qgc\_bg.qm</file>

<file alias="qgc\_de">resources/qm/qgc\_de.qm</file>

<file alias="qgc\_el">resources/qm/qgc\_el.qm</file>

<file alias="qgc\_en">resources/qm/qgc\_en.qm</file>

<file alias="qgc\_es\_ES">resources/qm/qgc\_es\_ES.qm</file>

<file alias="qgc\_fi">resources/qm/qgc\_fi.qm</file>

<file alias="qgc\_fr">resources/qm/qgc\_fr.qm</file>

<file alias="qgc\_he">resources/qm/qgc\_he.qm</file>

<file alias="qgc\_it">resources/qm/qgc\_it.qm</file>

<file alias="qgc\_ja">resources/qm/qgc\_ja.qm</file>

<file alias="qgc\_ko">resources/qm/qgc\_ko.qm</file>

<file alias="qgc\_nl">resources/qm/qgc\_nl.qm</file>

<file alias="qgc\_no">resources/qm/qgc\_no.qm</file>

<file alias="qgc\_pl">resources/qm/qgc\_pl.qm</file>

<file alias="qgc\_pt\_PT">resources/qm/qgc\_pt\_PT.qm</file>

<file alias="qgc\_ru">resources/qm/qgc\_ru.qm</file>

<file alias="qgc\_sv\_SE">resources/qm/qgc\_sv\_SE.qm</file>

<file alias="qgc\_tr">resources/qm/qgc\_tr.qm</file>

</qresource>



1. 在qt中打开Sources/src/ QGCApplication.cc

第464行改为：if(\_qgcTranslatorQtLibs.load(":/qm/qgc\_" + \_locale.name())) {

完成以上更改后，再进行编译运行，则汉化包可以使用。

1. 代码框架分析
2. main.cc文件

