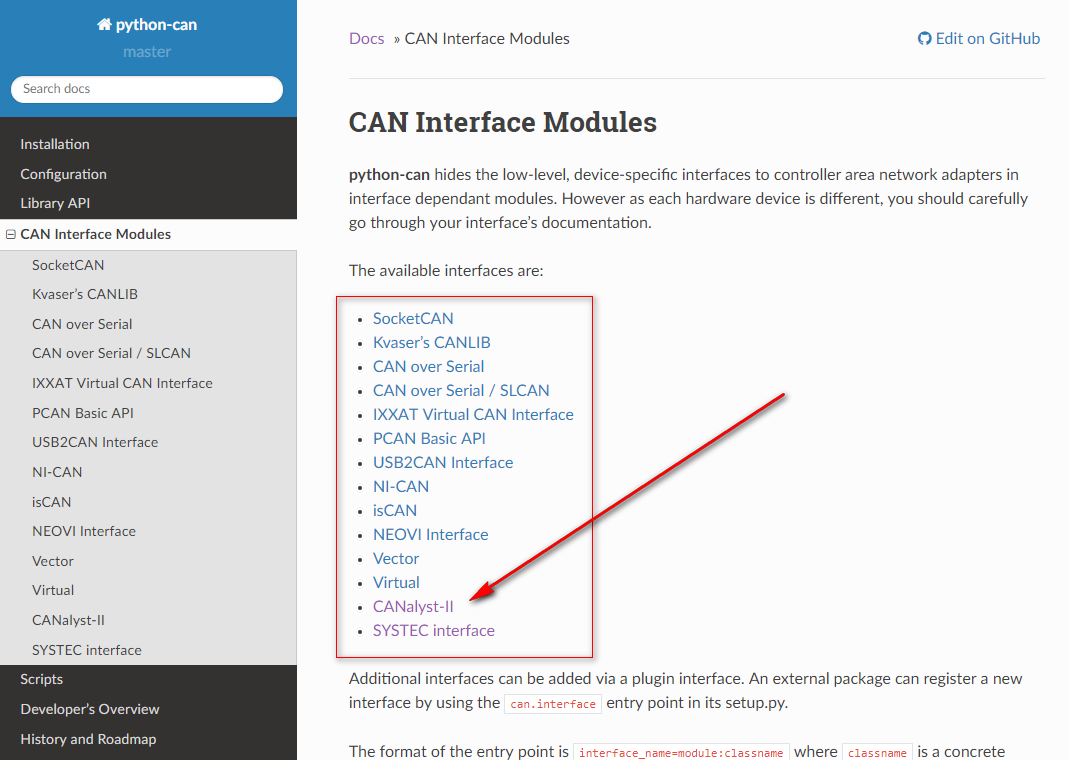
1、不推荐使用python-can库去开发我们的USBCAN设备

python-can实际对市面上主流的多种CAN设备进行了多层次封装，统一上层接口，以通过简单的设备型号指定，即可使用对应型号的CAN设备。这会带来一个问题：不同的CAN设备有不同的接口与功能，简单的进行多层次封装接口，使得单一的CAN设备配置不灵活，部分功能不能发挥。

下面这个列表就是python-can支持的CAN设备列表，列表中CANalyst-II就对应我们公司系列产品（所有产品都可以用该型号开发）。

<https://python-can.readthedocs.io/en/master/interfaces.html>



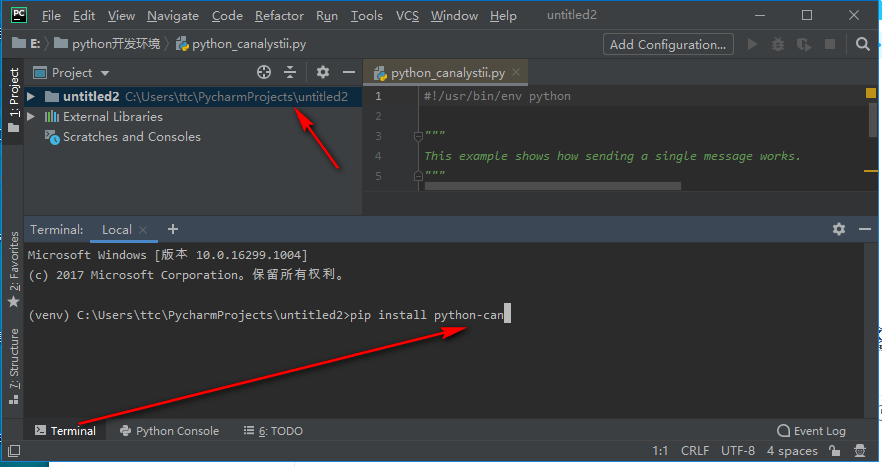
2、python实际上可以直接通过加载DLL库，实现对CAN卡的二次开发，我公司提供了对应的样例： CAN分析仪资料\二次开发示例源代码\python3.8.0。这种方法比python-can更简单，更有针对性，可以充分发挥CAN卡的功能。推荐使用！

3、python-can库，直接安装使用，会报错，我公司对python-can库中，canalystii.py文件进行了升级，使用时需要替换该文件。官方已识别该错误并更新，后期会在新的python-can版本中修复该BUG。

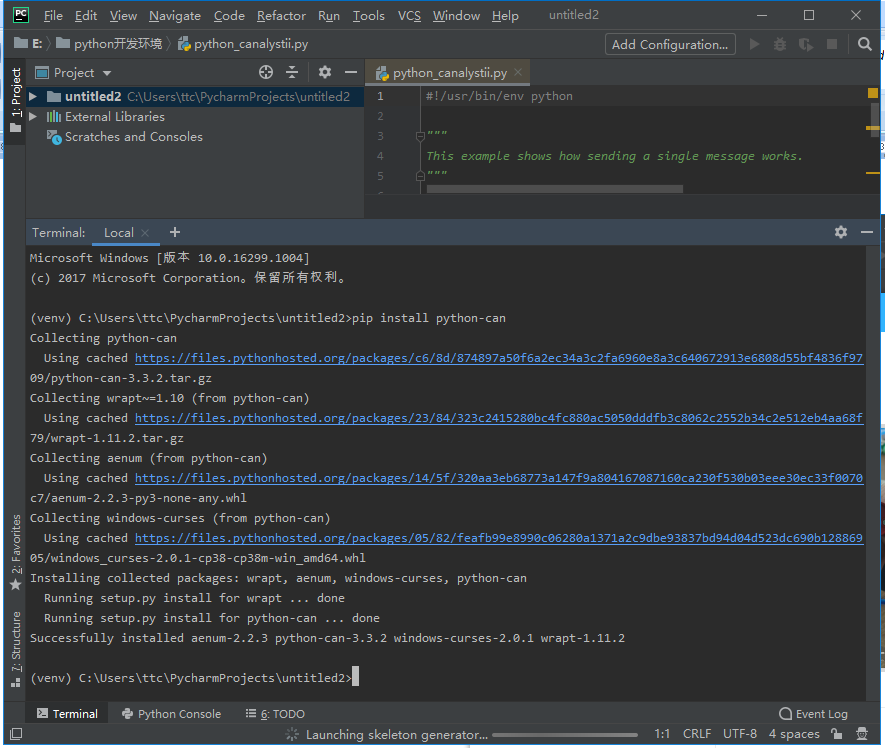
4、编程平台pycharm-community-2019.3.exe 64位。

5、新建工程、安装python-can库、打开样例。注意工程目录，python-can会安装在这个目录下。

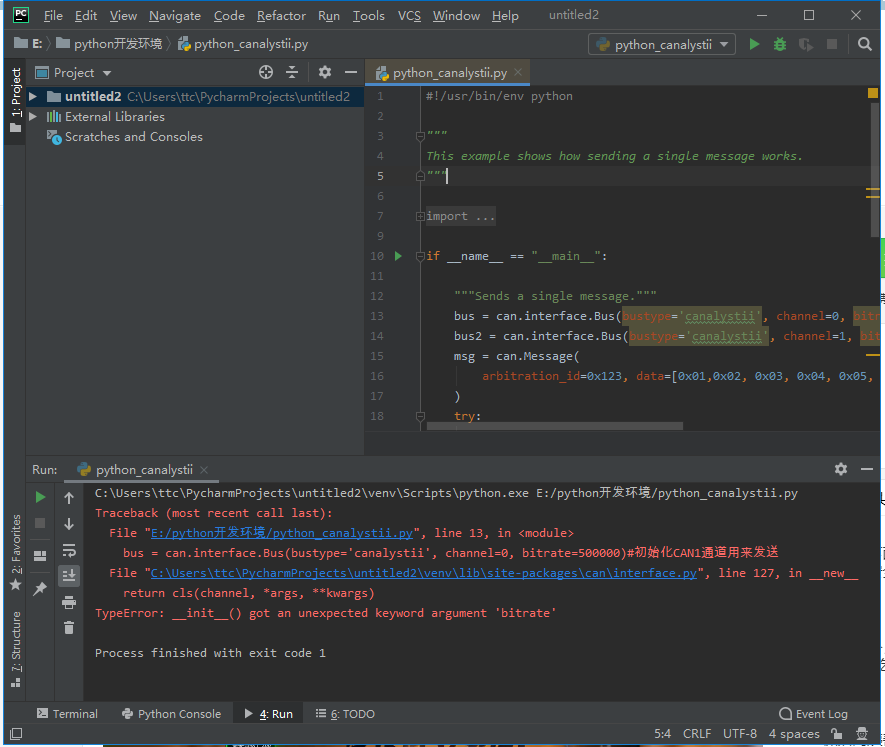
命令：pip install python-can回车



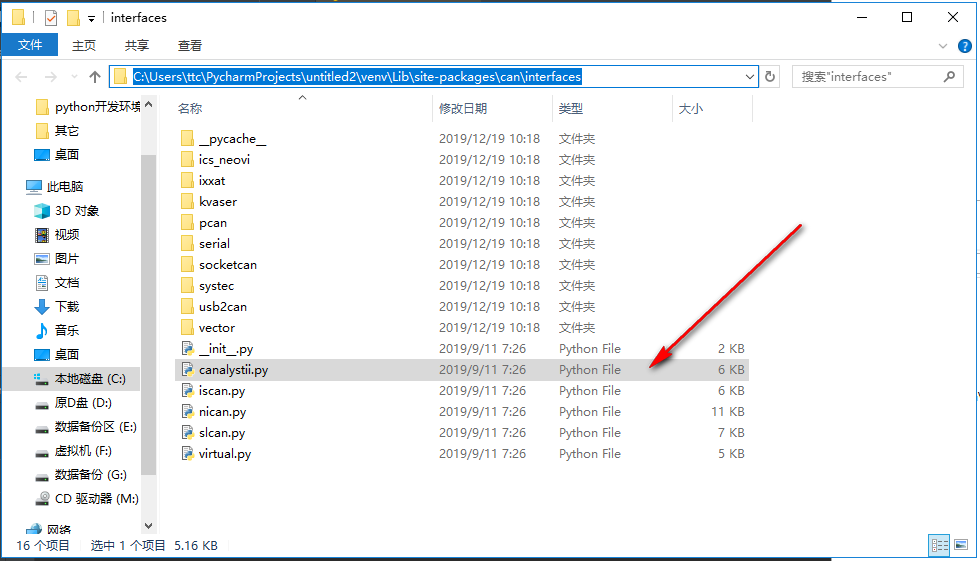
正确安装到工程目录：



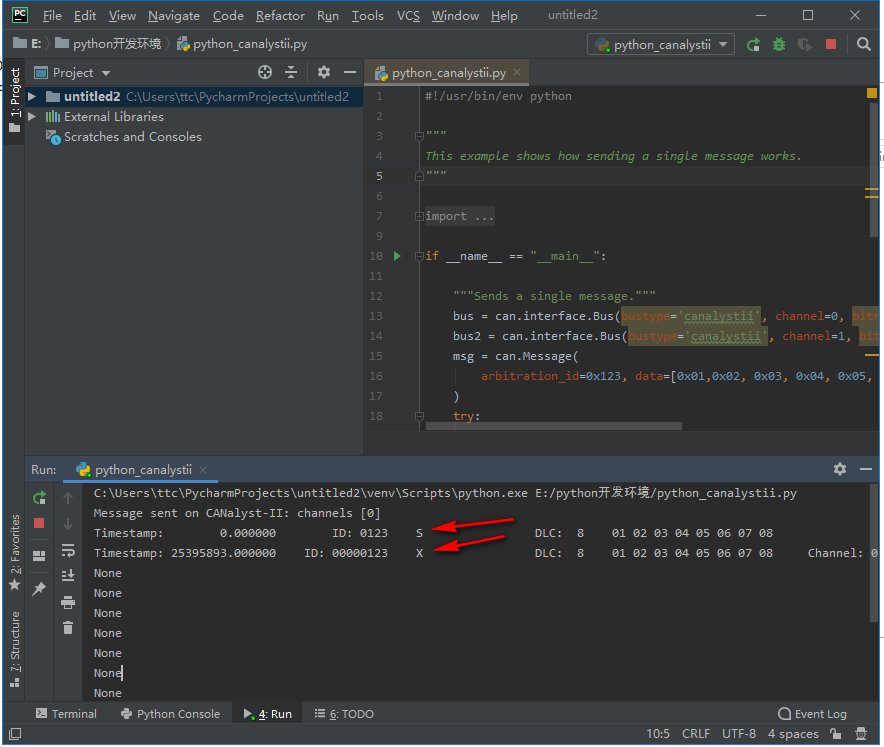
直接运行样例报错：



替换python-can库中，canalystii.py文件



再次运行样例，CAN1发送1帧，CAN2接收一帧。注意：把CAN1与CAN2对接。



6、ControlCAN.dll要与python\_canalystii.py放在同目录下，32位编译器用32位DLL。64位编译器用64位DLL