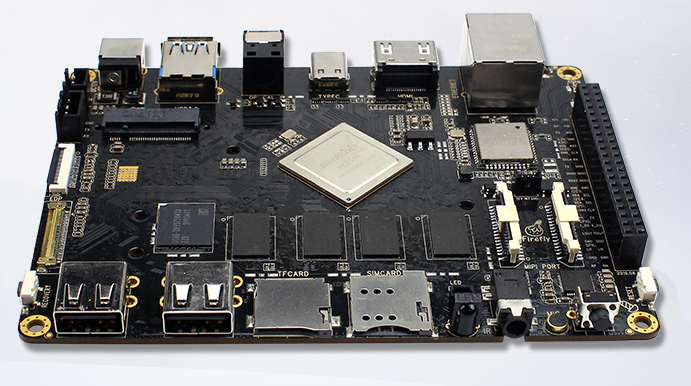
1. 安装firefly屏幕模组

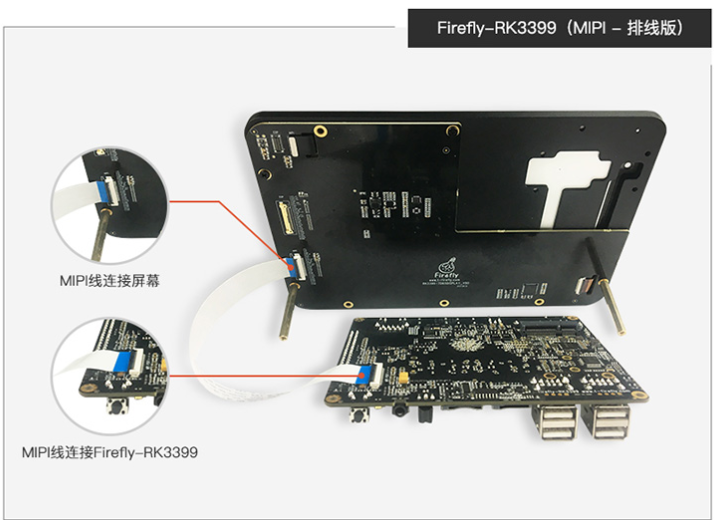
[http://www.t-firefly.com/doc/product/info/id/85.html#EDP7.85.E7.BC.96.E8.AF.91?tdsourcetag=s\_pcqq\_aiomsg](http://www.t-firefly.com/doc/product/info/id/85.html" \l "EDP7.85.E7.BC.96.E8.AF.91?tdsourcetag=s_pcqq_aiomsg)

（firefly-rk3399官网）

这个屏幕模组是与rk3399想匹配的模组，安装firefly-rk3399



firefly-rk3399



目的：安装这个屏幕模组是为了将rk3399板子的操作过程简化，目前实现人脸识别部分只能通过鼠标，键盘，外接显示器去操作，非技术人员安装时也比较复杂，如果可以直接使得firefly-rk3399直接连接屏幕就会方便的多。

安装过程;

<http://wiki.t-firefly.com/zh_CN/Firefly-RK3399/module_display.html>

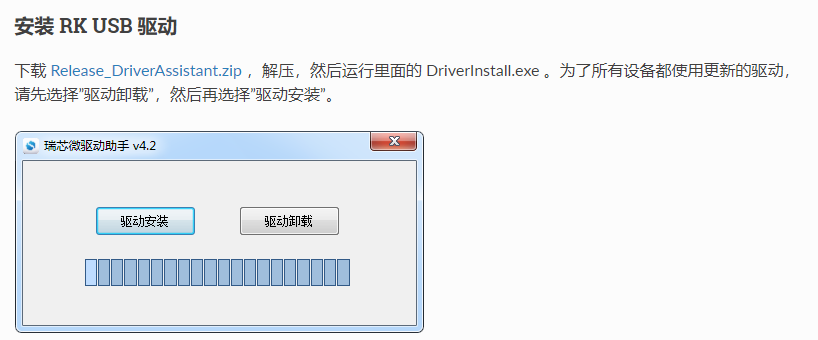
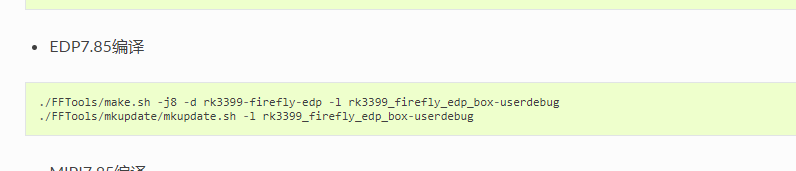
这里列出firefly-rk3399的官方技术文档。如果有解释的不详细的部分，可以看看文档。

屏幕模组有两款

我选择了edp屏幕模组，如果之后有直接支持hdmi接口，可以选择hdmi接口的，安装也比较方便。



选择edp屏的理由是，官方只支持两款屏幕edp和mipi的。

1. 按照官方文档推荐的方法连接edp屏幕
2. 要支持edp屏幕必须要安装firefly-rk3399的屏幕，这一部分可以按照启动模式中Loder模式进行操作，然后可以按照升级固件中的系统操作方式来安装，先安装usb驱动，然后下载edp屏幕映像。
3. 
4. 然后按照烧写方法去烧写升级固件就可以了，升级映像可以在donsok的百度云里面，至于文件可以跟邢明政老师要一下下载链接
5. 烧写固件时要求用linux命令烧写
6. 
7. 然后把这些文件统一打包成映像文件。烧写即可

注意点：烧写映像之前一定要把彭峰科技的板子中model\_data的数据拷贝出来，烧写数据之后系统完全还原，所有的东西都会清空，要不然还得把板子邮到彭峰科技重新烧写

我遇到的问题;烧写之后连接edp屏幕用原始相机无问题，但是使用人脸识别部分就会出现相机倒置，我猜测这里其实转移以下彭峰科技部分Camera的setDisplayOrientation的角度。

建议你申请一块板子先测试开始，我当时做到的部分是调试部分，显示相机正常已经可以识别人脸了，同时也连接了摄像头模组，但是进入rk3399的系统发现这里面没有返回键，如果要连接屏幕的话，可以实现，但是界面部分一定要添加返回这样的导向按钮，不然没有办法控制。

1. Git使用，代码管理

公司中使用的gogs网址<http://10.3.1.165:3000/user/login>

如果之前有svn和git基础的可以直接看下面两篇技术文档

<https://www.bootcss.com/p/git-guide/>

<https://cloud.tencent.com/developer/article/1355182>

Git主要用于公司代码管理，可以从远端仓库获取公司正在开发部分的代码，同时提交你自己的代码，协同开发

进入gogs页面之后先注册



注册成功之后可以新建一个自己的仓库（创建成功之后会有这样的显示：）

你可以先在本地仓库

Mkdir gogsworspace

Cd gogsworkspace

Git init

新增文件

Touch readMe.Md

Git add .

Git commit -m “这里写你要提交部分的英文注释，添加，删除，修复功能之类的”

Git remote add origin 你的仓库

Git push origin master

开发过程中，我们实际上使用的git指令并不多，如果有使用困难的话，推荐可以看一下git操作文档<https://www.w3cschool.cn/isrekq/em9dzcqm.html>

我在操作过程最常用的指令为

Git clone 远端仓库

更改完你所有部分的代码之后

Git pull

Git push

如果中间过程中出现cerfication deny你可以使用git config --global命令

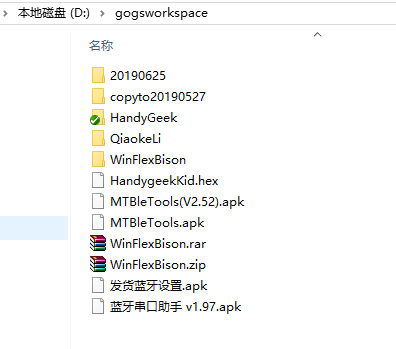
添加user.name,user.email等操作

如果需要看提交记录的话可以直接使用git log

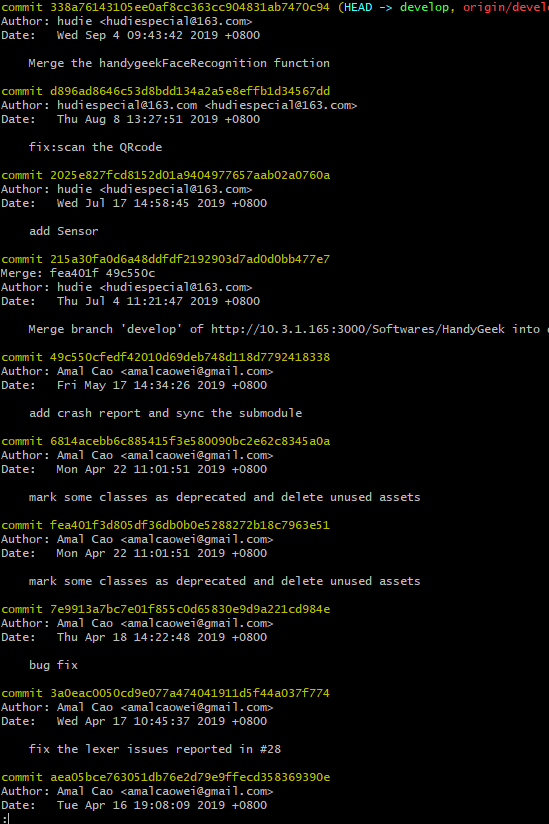
Git checkout切换分支

查看代码区分git diff

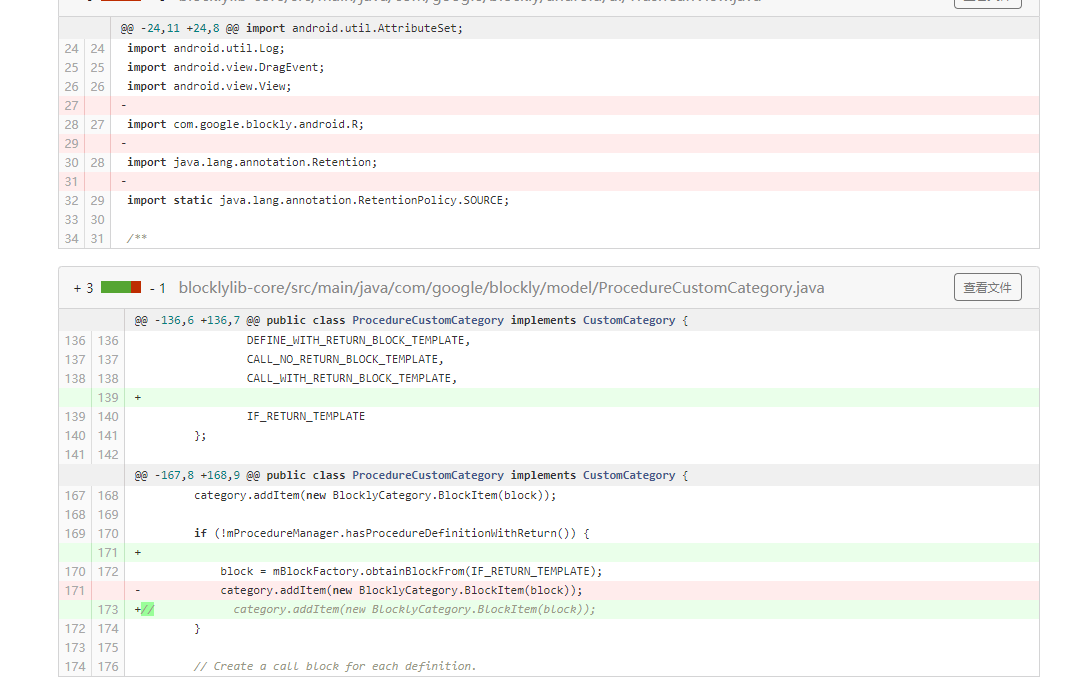
也可以使用版本回退，git reset --hard 这个指令我只测试过提交代码时候并没有用过，更简单的方法可以多备份几次你的代码



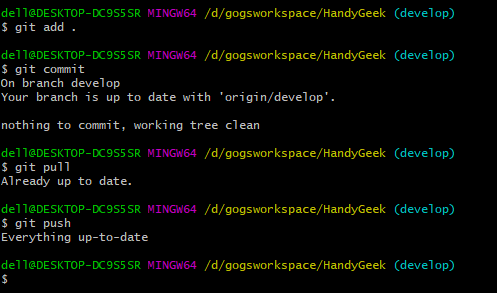
其中动手客代码在HandyGeek里面，同时git上也有提交的一份



这里面是提交记录，每个节点都可返回，或是查看更改添加的部分

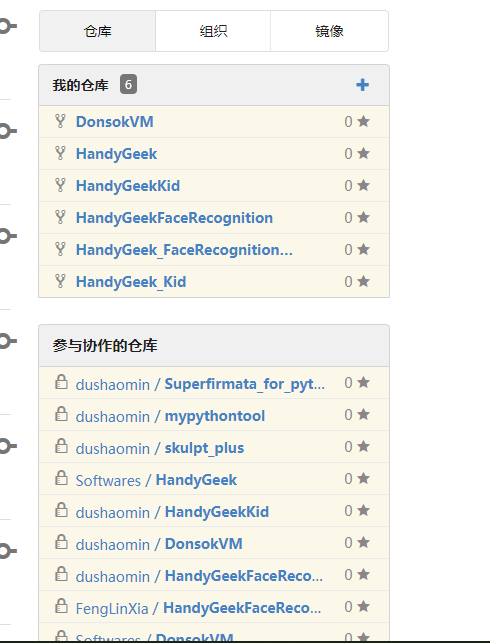


通过颜色区分可以查看更改添加删除的部分



提交流程大致如上

登陆gogs之后协同的仓库还有你的仓库都在控制面板主页



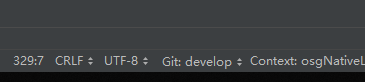
如果需要代码的话，需要向杜绍敏老师申请，我的代码全部在D盘下的gogsworkspace下面

其中HandyGeek是动手客，QiaokeLi部分是巧颗粒，20190625部分是最新的人脸识别

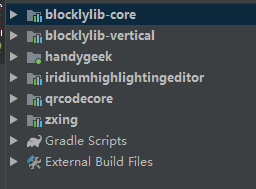
Copyto20190527是之前为了做出内部人员测试部分所做的demo，NewFaceDemo是和彭峰科技那边对接的库文件，蓝牙模块里面有之前馒头科技的资料，Donsk.keystore是handygeek的签名文件，这个签名文件需要你熟悉代码，且能够进行开发之后向邢明政老师作出声明，你已经拿到了动手客的签名，请务必妥善保管。

1. HandyGeek代码

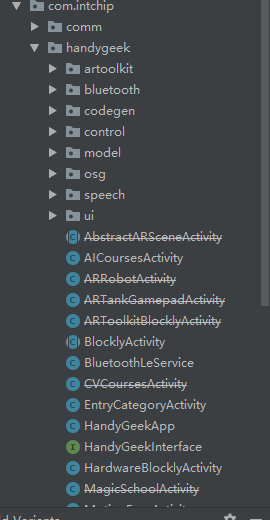
HandyGeek代码有一个子库DonskVM，其中DonskVM对应的是动手客atmega328芯片（白板子）所对应的下位机代码。我们现在开发的分支在develop分支，clone代码之后可以切换到develop分支，master分支是应用于之前四川aphla发货对应的版本。Develop分支是现在正在使用的版本。



AndroidStudio有直接切换分支的部分，可以直接切换，切换分支之前要保证代码分支之前的代码commit，一般情况下我们不对master代码进行开发。

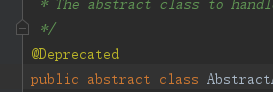


如上可以看到Blockly的库，编译器高亮的库，QRcode库ZXing库



主要用到类文件都在Android--》java文件中，可以做一些对于页面和处理逻辑的更改，可以从SplashActivity中进入及界面进行调试。

有些现已不用的代码已经被废弃了



其中AICoursesActivity是用于巧颗粒的登入，BlocklyActivity是Blockly核心代码的实现，BluetoothLeService是蓝牙GattSever和给于的BluetoothLEdevice中的连接。

EntryCategoryActivity

是巧颗粒入口选择课程的界面

HardwareBlocklyActivity

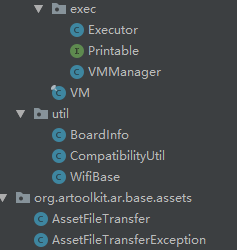
是HandyGeek进入后的主页面，其中有Blockly块的注册，添加外设部分。

QRCodeScannerActivity

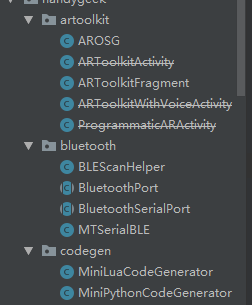
二维码扫描界面

SourceCodeActivity

Python工具进入页面

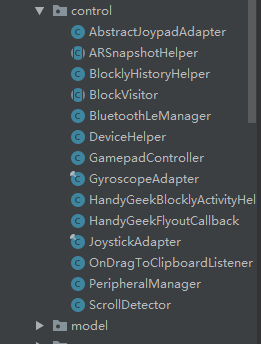


这一部分建议和DonskVM联系在一起阅读，在有了大致逻辑的梳理后可以尝试着自己更改一些块的实现。

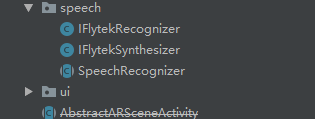


这一部分是AR中图像识别，蓝牙部分，codegen生成lua,python

代码

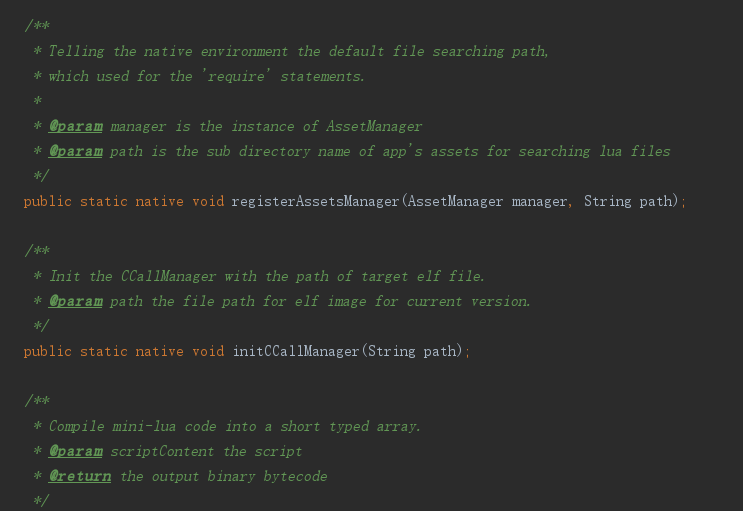


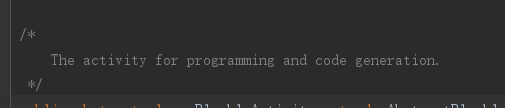
这些类主要是gamepad 图像识别，blockly,蓝牙管理器，设备管理器，（gamepad）界面，拖动事件，外设管理器，滚动检测的实现



语音部分-->讯飞科技实现

以上所列出的都是项目介绍，具体的细节部分可以看代码部分，每个地方都有代码注释和模块介绍。



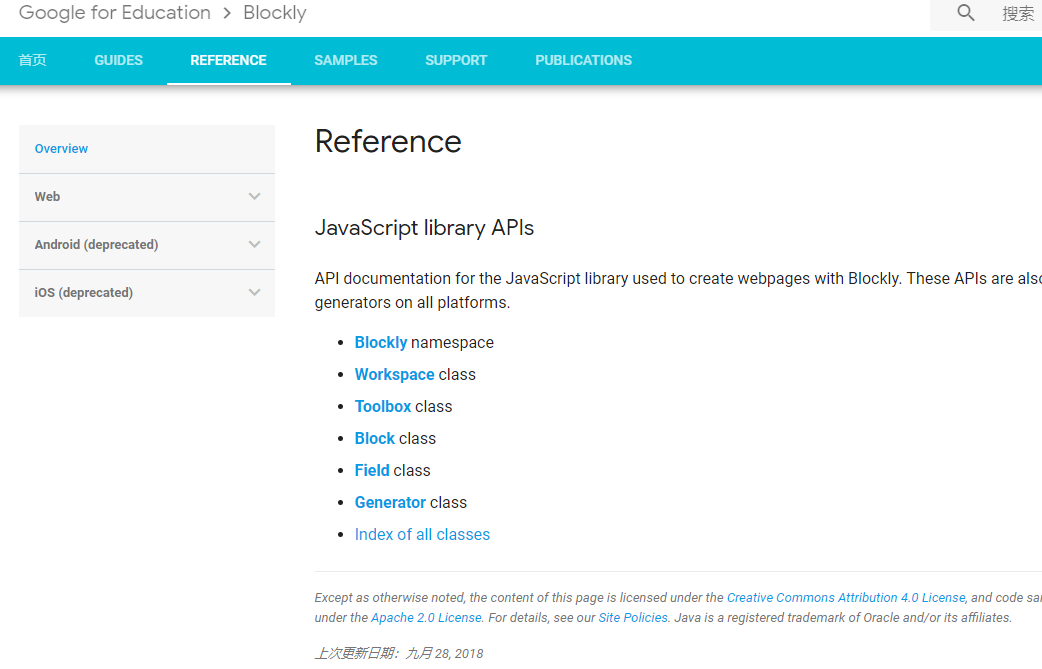


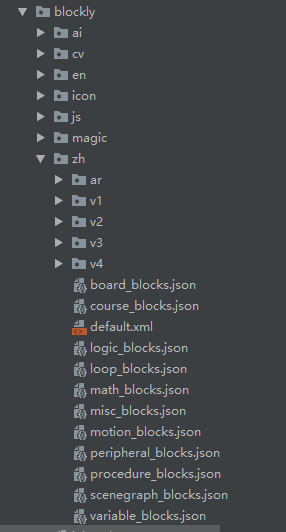
如果为了便于理解，可以在代码上加注释，尽量英文，如果难度很大，就可以先给自己要做的模块加注释，之后代码提交的时候再把注释部分改成英文或者无意义的log直接删除即可。

读懂Blockly块的前提是要把Blockly基础读懂，

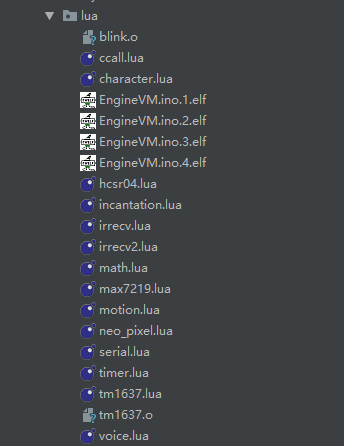
<https://blockly-demo.appspot.com/static/demos/index.html>

推荐你尝试看一下Blockly基础知识，对于更改代码部分上手的话比较快。

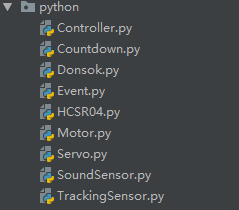




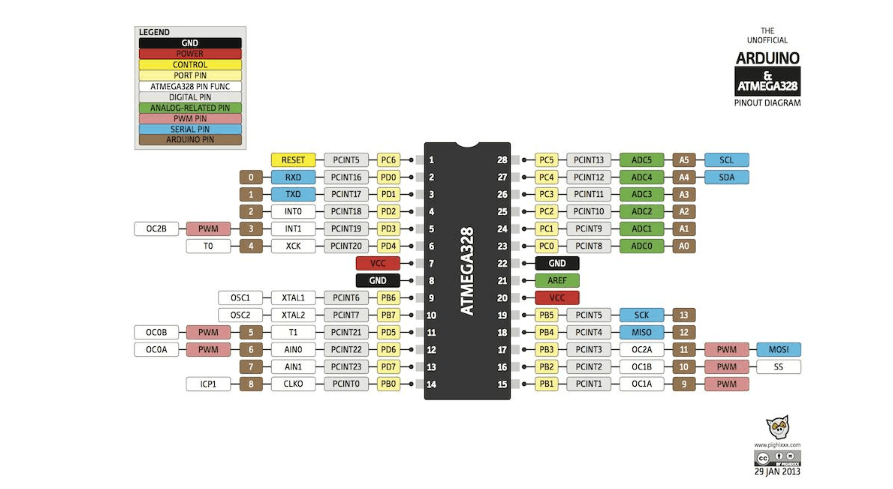
其中blockly块定义在这里



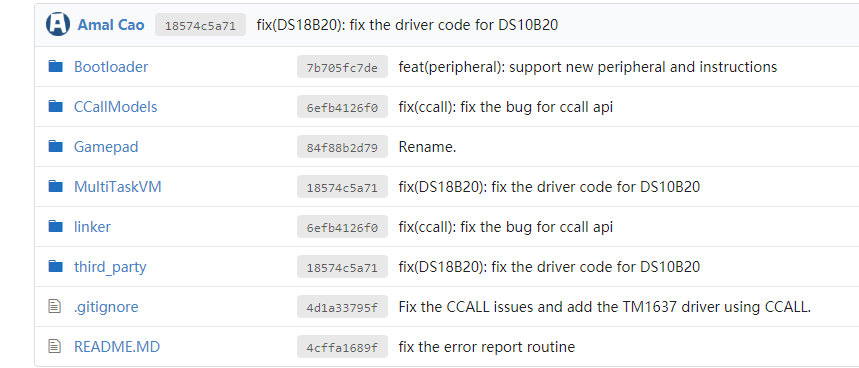
这里是lua代码定义，但是大部分实现的外设部分还是在DonskVM实现的



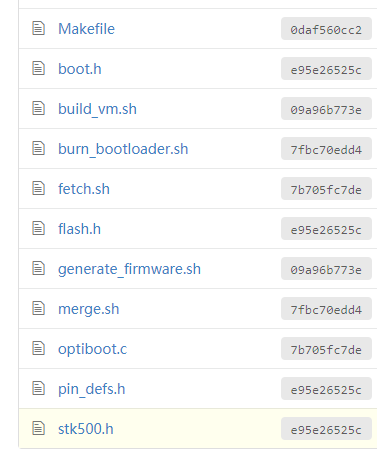
Python部分可以看一下常量值，这部分的定义可以对应



这一部分在看DonskVM中有实际的操作



DonskVM部分中Bootloader部分中boot启动问题



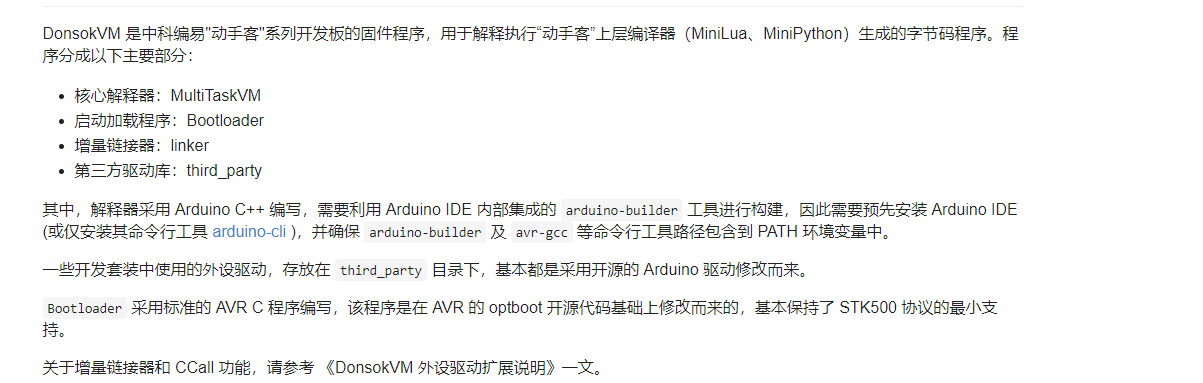
可以从generate\_firmware.sh看起，同时阅读build\_vm.sh,burn\_bootloader.sh,这些主要是利用Arduino生成的firmware.4.hex,便于烧写进入下位机，flash.h指的是烧入程序地址，可以通过这一部分查看烧写位置，bootloader的程序更改主要有下位机硬件工程师更改，所以这部分代码可以了解即可

CCallModel模块可以参考

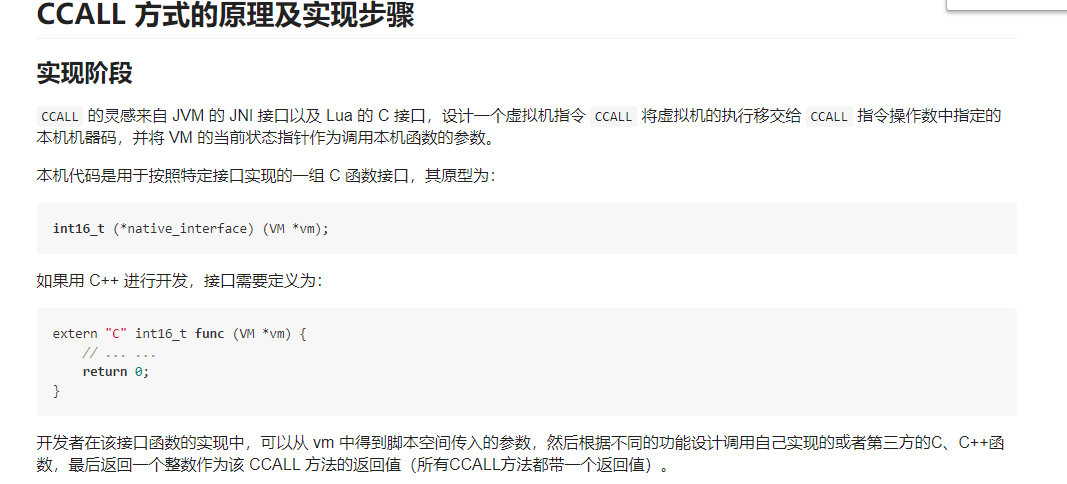
<http://10.3.1.165:3000/Softwares/DonsokVM/wiki/DonsokVM+%E5%A4%96%E8%AE%BE%E9%A9%B1%E5%8A%A8%E6%89%A9%E5%B1%95%E8%AF%B4%E6%98%8E>

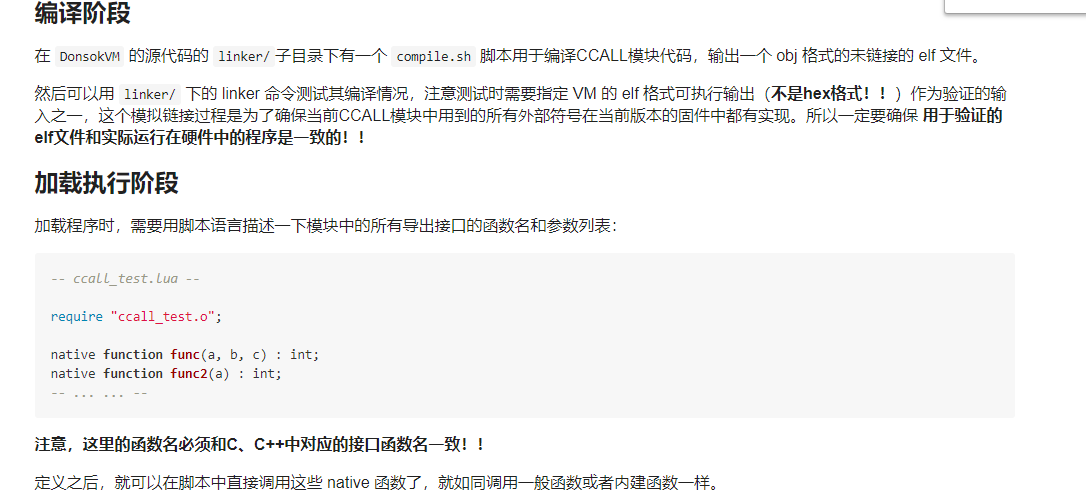
<http://10.3.1.165:3000/Softwares/DonsokVM/wiki/DonsokVM+%E7%BC%96%E8%AF%91%E6%9E%84%E5%BB%BA%E8%AF%B4%E6%98%8E>

里面都有对DonskVM的详细介绍







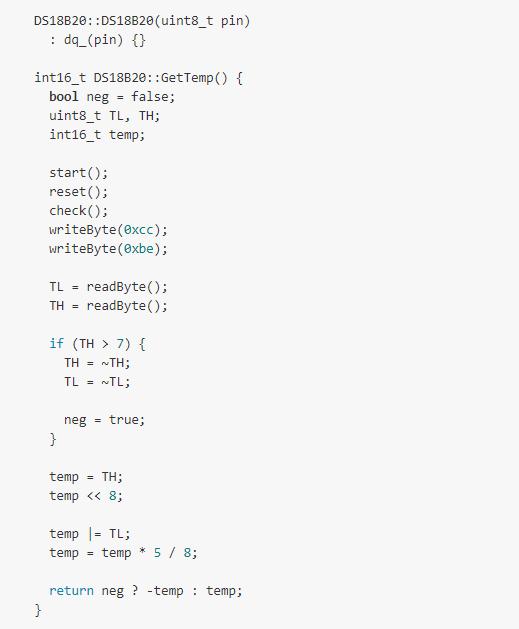


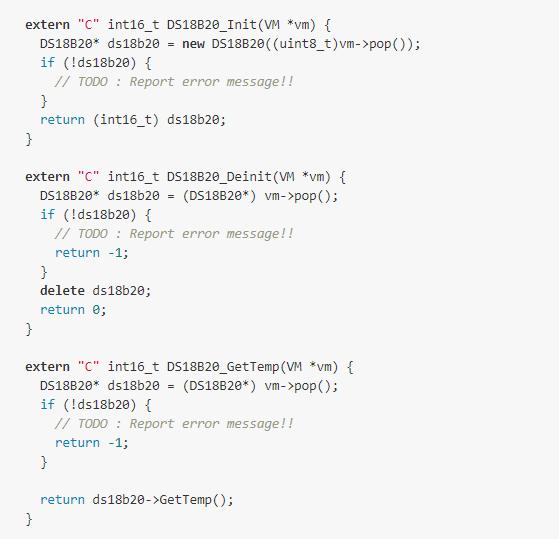






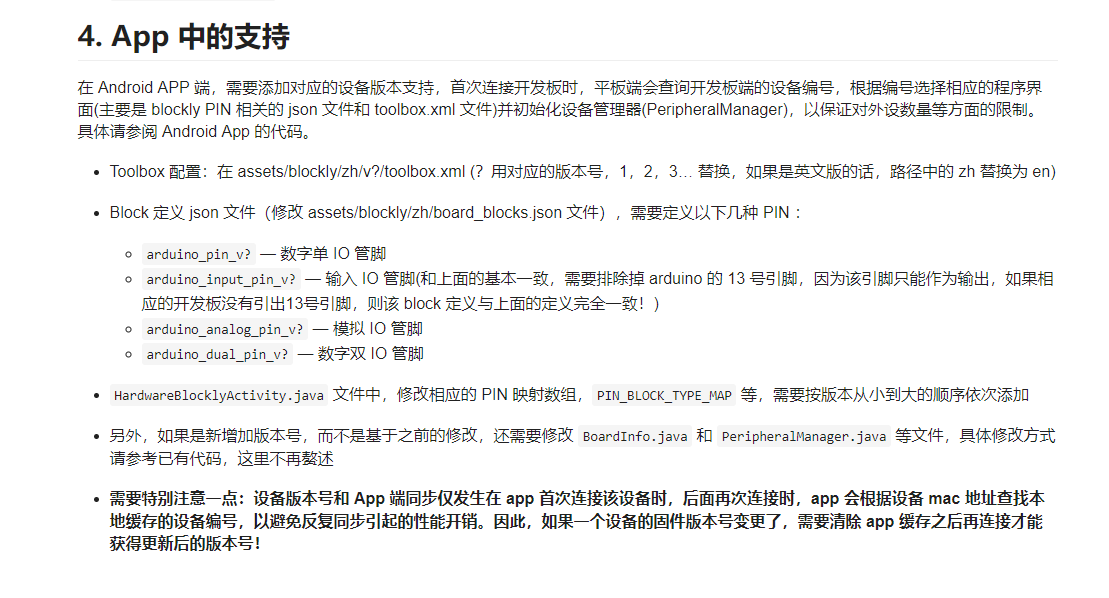




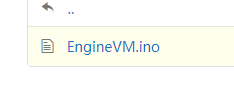


阅读这部分代码进行理解





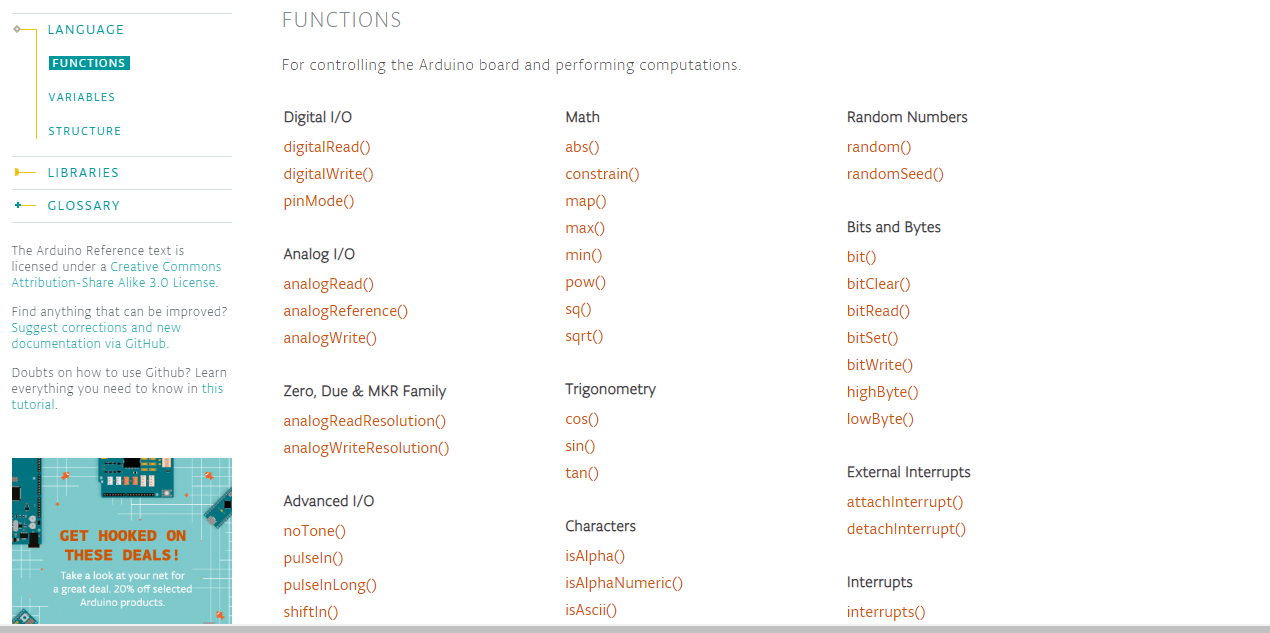
其中MultiTaskVM中VM.hVM.cpp,VMManager.h,VMmanager.cpp中要仔细理解。可以多花些时间看一下。

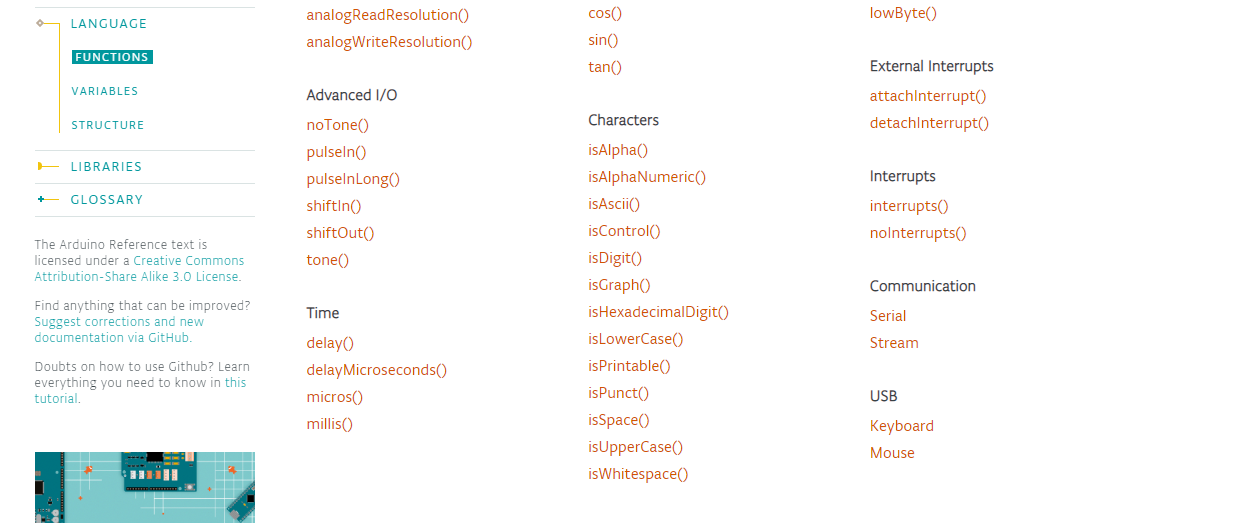


这一部分请参考Arduino的工作方式看一下。

对Arduino的理解部分，建议你学习以下Arduino语言对于硬件层代码理解有很大的帮助。

如何学习Arduino





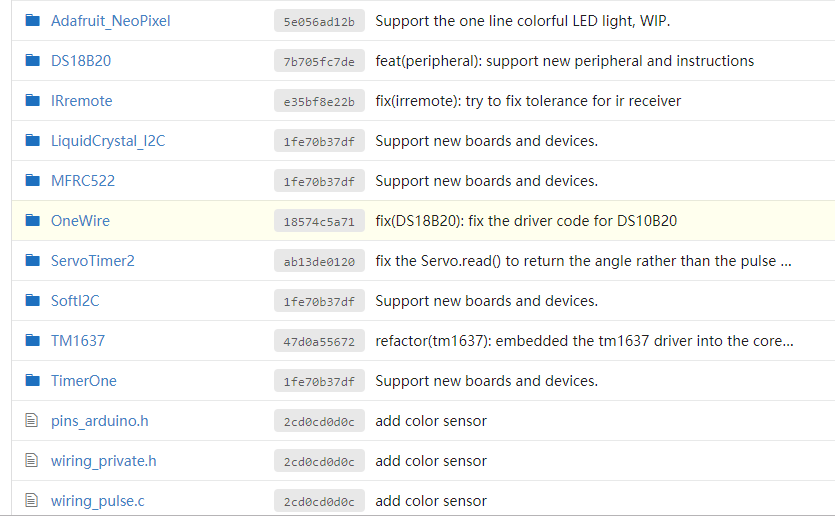
可以阅读以下Arduino基础函数的语义，然后找一本Arduino入门参考书理解以下大致函数的使用。

同时理解VM也建议你找一本有关编译原理的书去简单了解一下。

公司书架上就有很多可以用的到的书，你可以尝试理解以下。

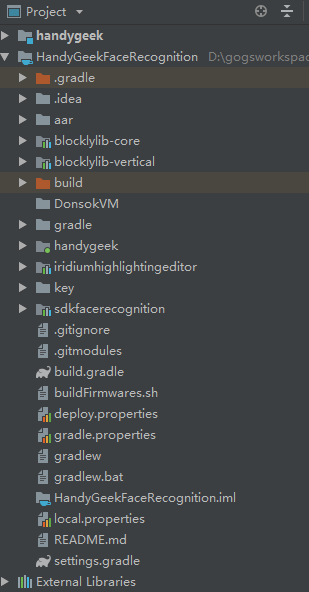
编译原理不推荐你找网上的课程一点点的看，非常的浪费时间，学习效果还不是很好。

Third\_party部分

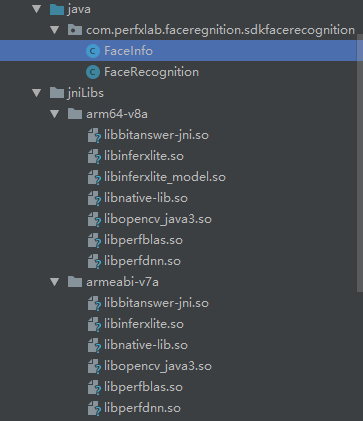


这里面包含了第三方软件库，如果之后想要增加部分外设，可以先把库文件添加到这里面，然后再对你要添加的硬件模块进行开发。

HandyGeekFaceRecognition



handygeekFacerecognition人脸识别部分大致和handygeek项目结构一样，所以这一部分只针对部分文件做出解释。



faceInfo是人脸的类。Facerecognition中init是嵌入识别的库文件，打开识别线程进行分析，getFeature通过调用库中nativeGetFeature方法进行识别，

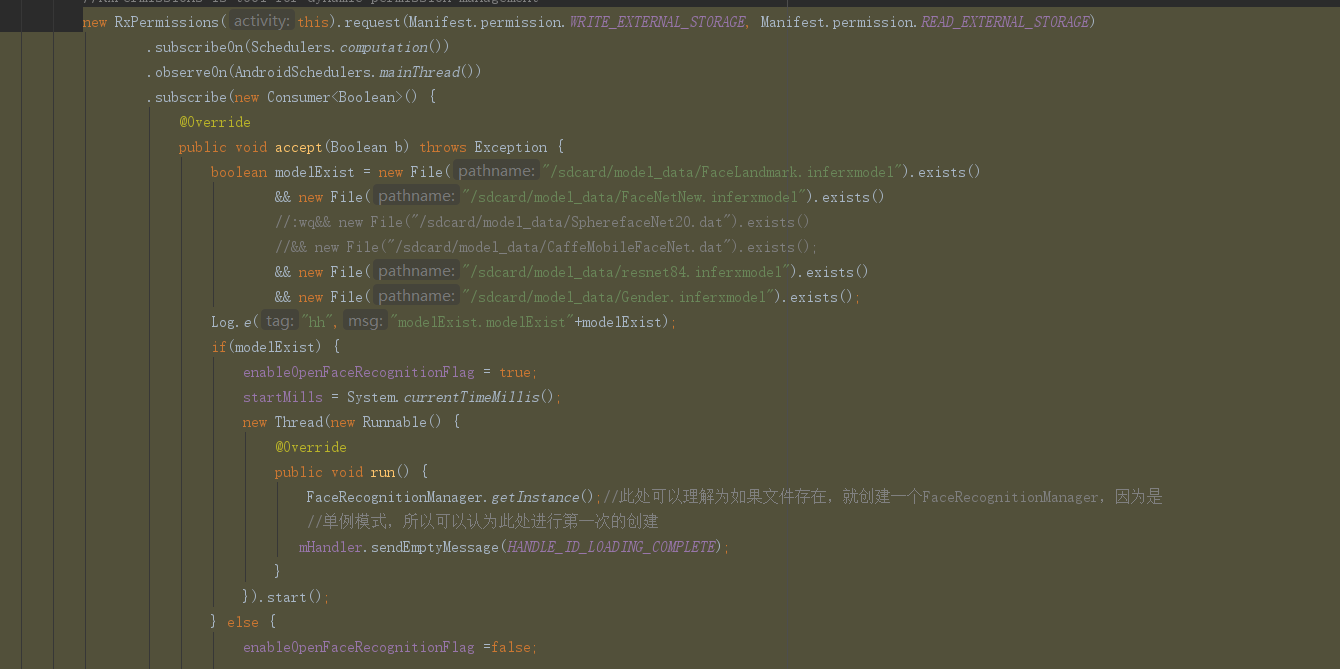
同理还是有getScore方法

Arm64-v8a里面是64位处理器里面的库文件，如果彭峰科技人员那边更新了，替换的库文件也从这里直接替换做二次匹配即可。

人脸识别的代码进入之后会有FaceRecognitionActivity

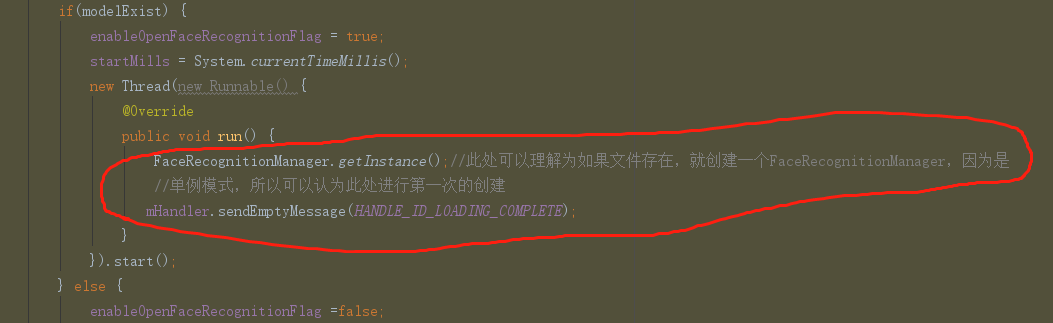
这个页面继承自HardwareBlocklyActivity通过onCreate方法



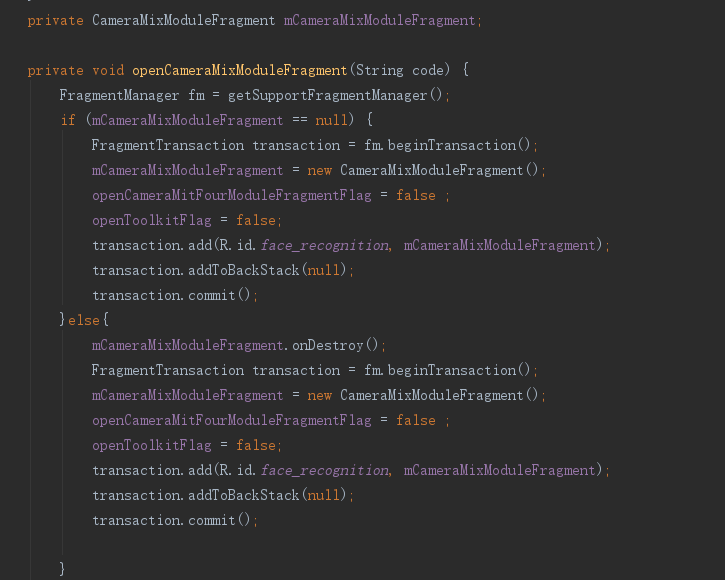


通过对accept

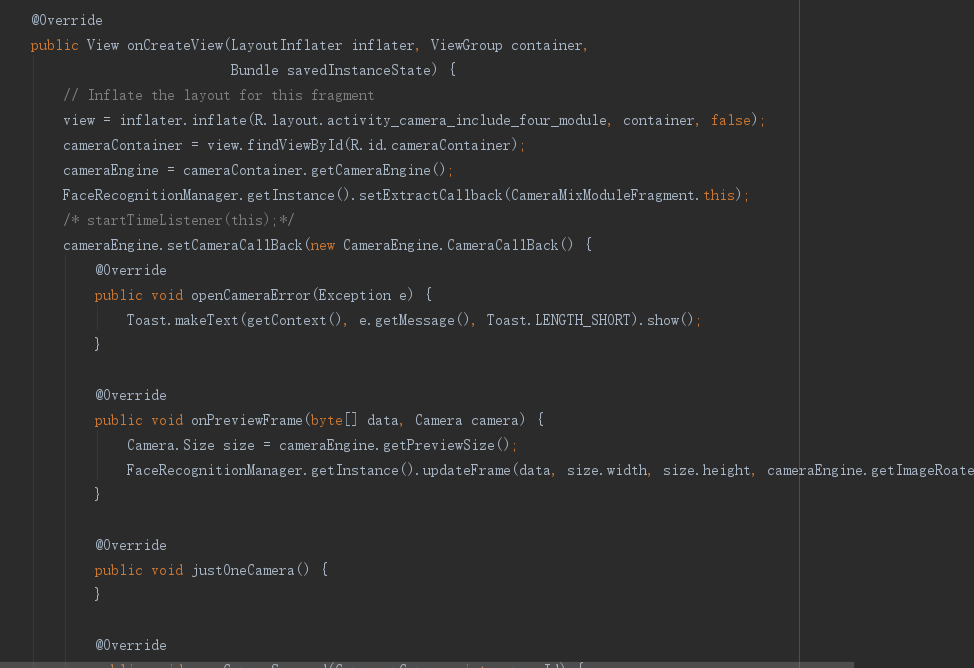
方法的实现去调用库文件里面的方法



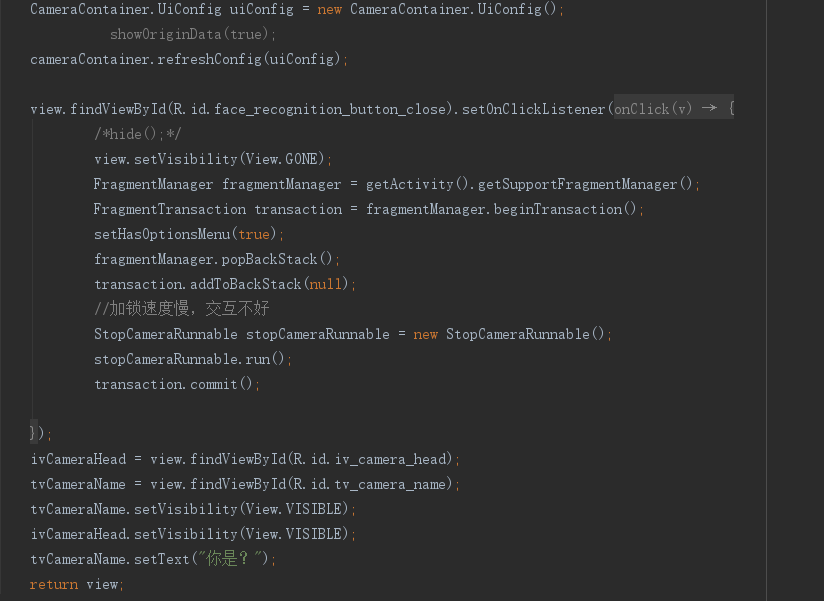
加载库文件创建界面之后，然后就是对方法的实现，原来的Activity变为fragment实现，这样的好处是activity与activity跳转之间会失去当前页面某些信息，但是activity跳转到fragment时不会发生种情况，这样做的好处是，点击连接蓝牙，跳转蓝牙不断开，这样用户在使用时会不需要重连蓝牙，同时这样也会比较稳定。



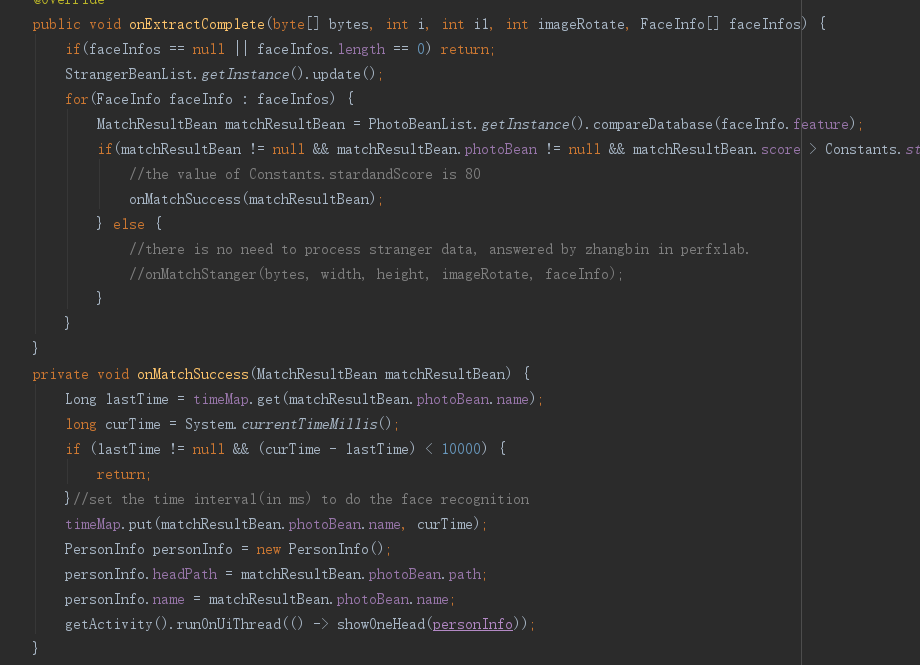
进入人脸识别界面

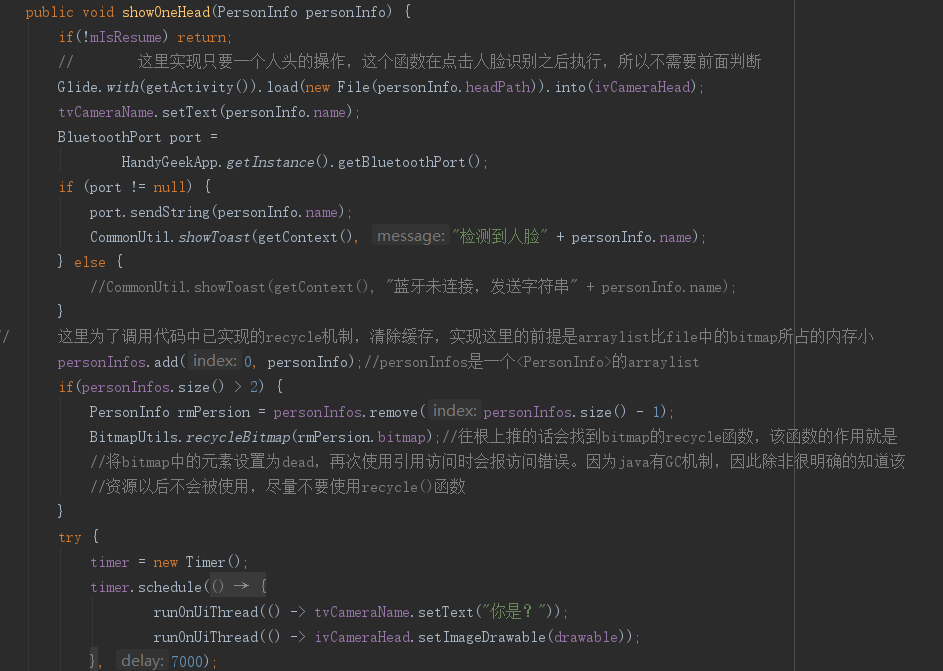


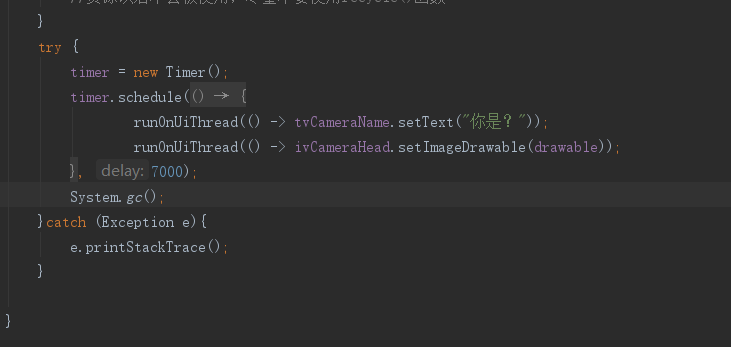
打开相机进行识别



识别方法主要在







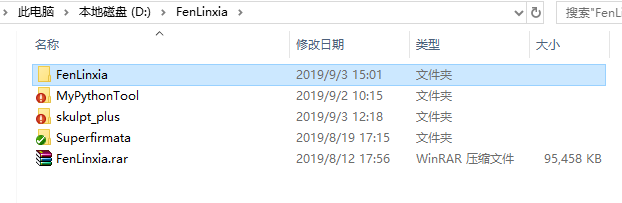
了解onMatchSuccess onExtractComplete

方法即可

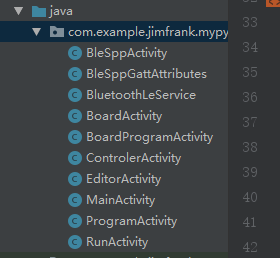
这部分之前的改动主要在showOneHead

里面如果后期界面需要优化更新，建议这一部分看懂并理解

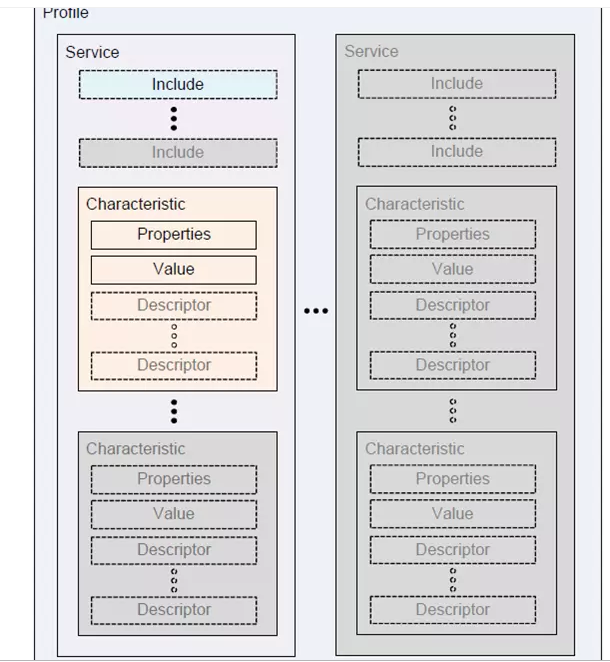
MyPythonTool:



这一部分所有的上位机代码和下位机代码全部在这里



读懂这个项目的前提是对蓝牙部分有一定的了解







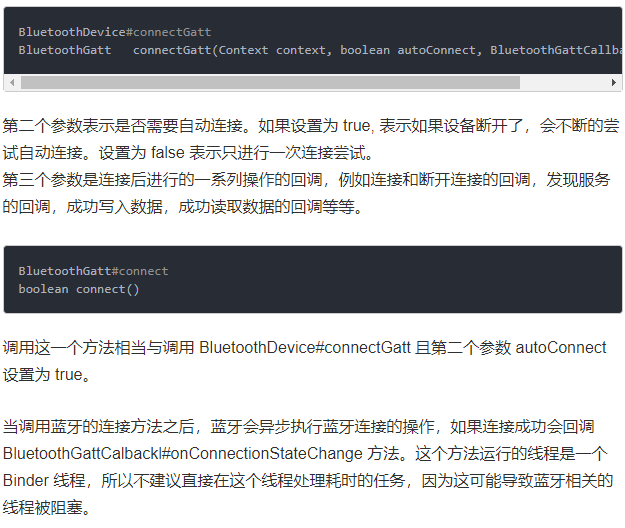












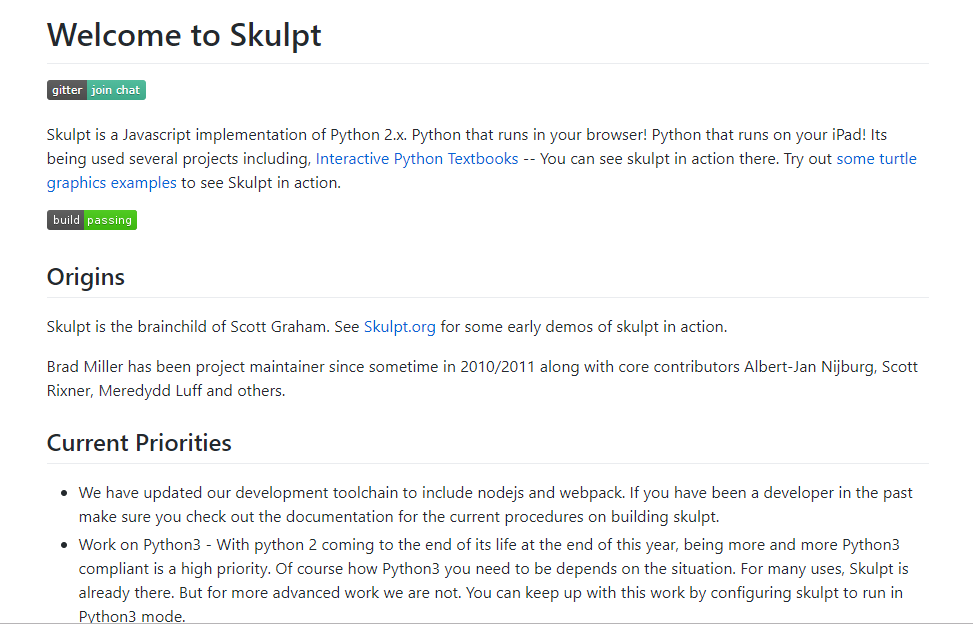
<https://www.yiibai.com/android/android_bluetooth.html>

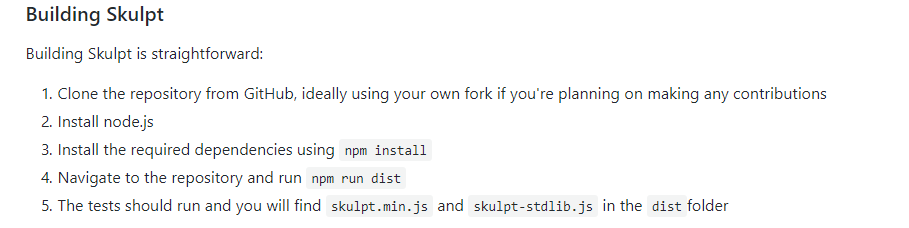
<http://c.biancheng.net/view/3169.html>

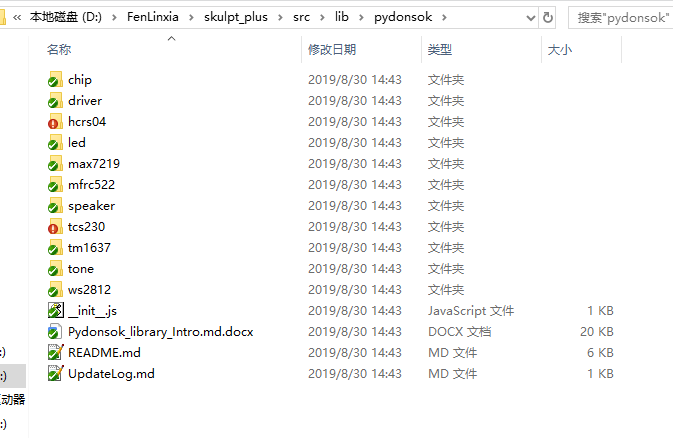
<https://www.jianshu.com/p/3a372af38103>

中间skulpt的操作

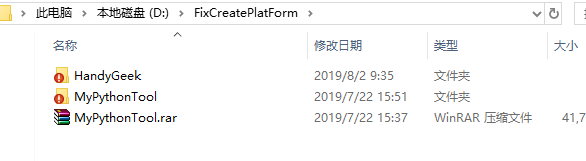
<https://github.com/skulpt/skulpt>







这里面只有hcrs04和tcs230需要再确认一下，其他模块需要测试一下功能的使用即可



是之前为了写两个项目整合成一个项目，做成一个平台的样式，这一部分也可以参考这篇文章进行理解

<https://blog.csdn.net/qq_33544860/article/details/51149560>

龙芯demo：

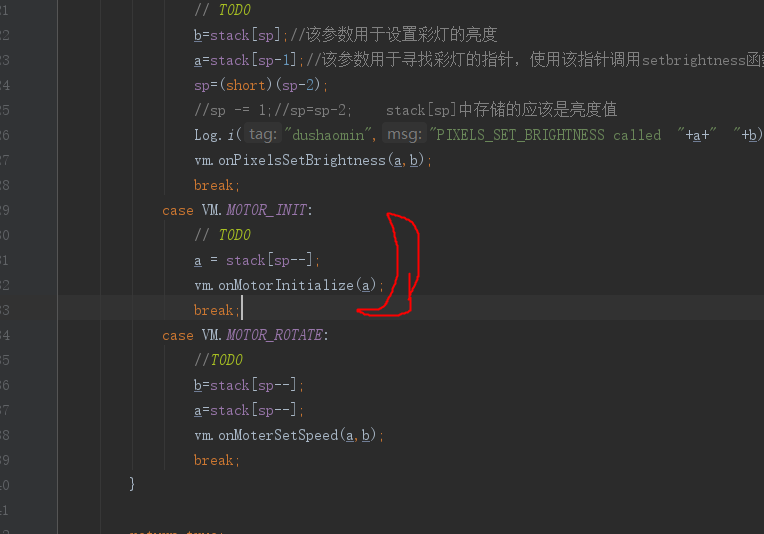
与龙芯合作部分主要是应用我们上层的apk，因此在实现过程中我们只需要提供对应的发送字符就可以，龙芯那边的工程师会根据我们这边提供的下位机代码进行更改，主要是遵守Firmata协议格式，对超长蓝牙字符传送进行中断处理。

Motor.initialize(flag)

其中flag为1表示为为了龙芯准备的版本，flag等于0为原来的版本



这四个为龙芯的引脚

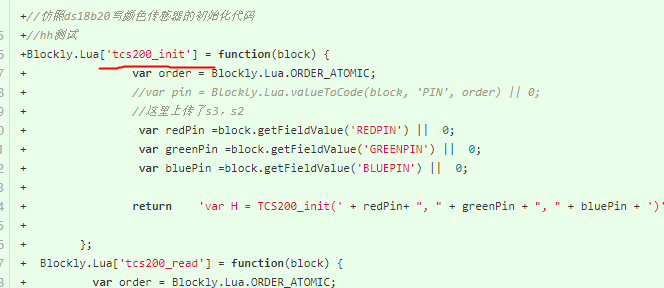


修改栈的弹出参数，使之适应flag，取栈顶元素作为flag



这两个要去掉以免给龙芯发送过多的无用字符，本来虚拟手柄的字符发送到pad上的虚拟机就行了，没有必要发送到下位机

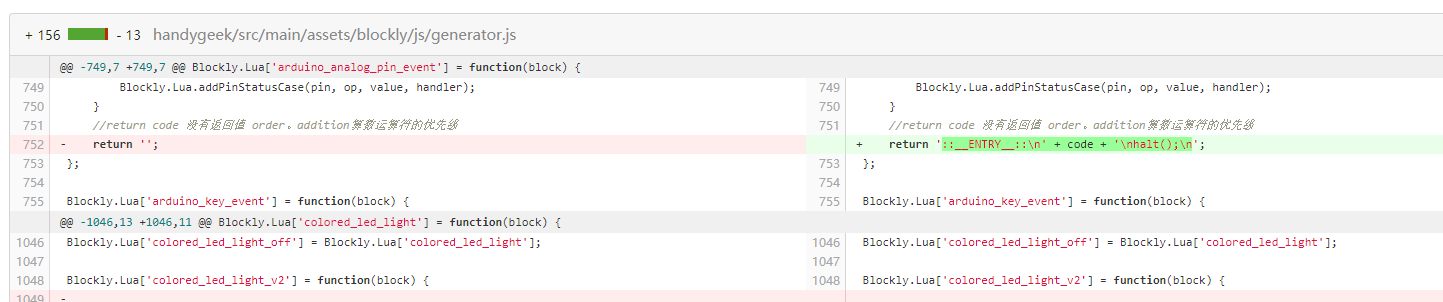
人脸识别部分代码详解：



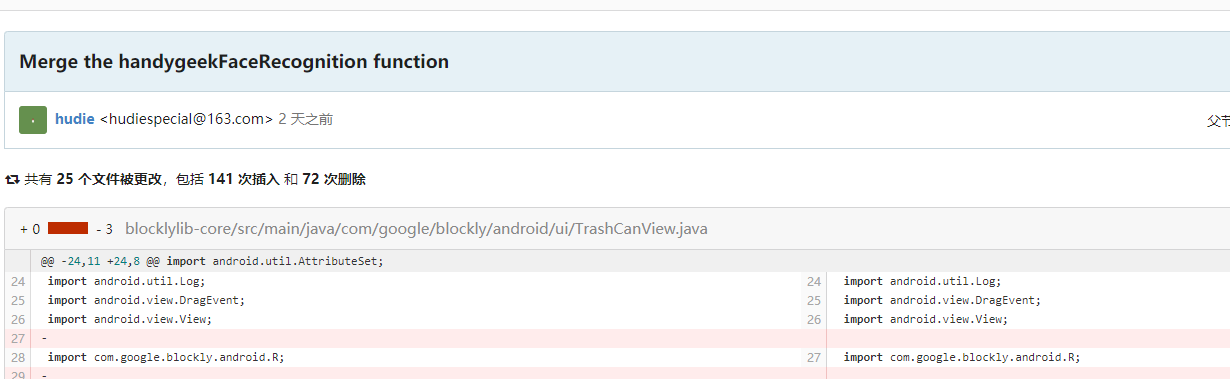
颜色传感器



台灯



这里的代码处理主要是因为只有gamepad时，无法直接运行代码中新添加的代码块，所以我在这里加入了起始代码段，这样在没有程序开始后这样的块，也可以使程序正常识别并执行下去。



这个节点实现表情界面，和视觉识别界面可以同时存在在一个主页面里，同时正对在键盘块中嵌入语音识别，表情块不会使程序无反应，在语音块中嵌入表情块和其他块可以正常执行。这一节点的实现可以参照HandyGeekFaceRecogniton之后添加的功能实现。



为了新增颜色传感器的块使用，所以要新添加共功能中实现的函数，而这个函数以来于这三个库文件，需要把他们新添加到第三方库中去实现。

颜色传感器：

因为颜色传感器一定要使用动手客白板子中的GN口的块，但是我们时用颜色传感器块获取颜色传感器的值无法在界面上显示，所以我们需要tm1637的块显示一下获取的值是不是跟获取的值是一样的，所以就要在写程序的时候先把所有颜色传感器的块连接好，等程序上传完毕之后，间隔一两秒等到颜色传感器开始识别之后，再开始利用tm1637的块识别

--祝工作顺利