经验总结报告

在这个项目中，我主要负责内核、文件系统的定制，摄像头驱动配置，QT 的移植。

项目刚开始，我们收集资料，确定使用 UVC 摄像头，因为 UVC 摄像头驱动在 Linux 内核中已经提供了。对于 GUI 显示和视频采集，我们找到了三种解决方案：

方案一：使用 QT4 写 GUI，直接调用 v4l2 接口读取摄像头数据

优点：最终生成软件体积小，不过调用 v4l2 接口读取数据很繁琐，容易出错。

方案二：使用 QT4 写 GUI，使用 opencv 处理摄像头数据

优点：处理数据简单，opencv 移植略为繁琐，软件体积较大。

方案三：直接使用 QT5

优点：QT5 在 multimedia 中有 QCamera 类，可以直接使用摄像头。

缺点：QT5 移植繁琐（后来证实为大坑）

一开始为了编码方便，采用了方案三，进行 QT5 的移植工作，中途发现很多问题。我们小组尝试了从 QT5.0 编译到了 QT5.5，发现很多错误。

1. QT5.1 及以下版本，在编译 qtwebkit 时候，找不到一些与 unicode 相关的 tou 文件(udat.h 等)，从网上下载 unicode 头文件，链接时出错。

2. QT5.2、QT5.3 版本，编译 quick 模块失败，无法解决。

3. QT5.4 及以上，开发板配备的交叉编译工具太老，不支持 c++11，编译到最后链接失败，需要更换交叉编译工具

我们小组主攻 QT5.5.0 版本的编译，主要操作操作如下：

下载最新的交叉编译工具(老版本会链接失败)

```
sudo apt-get install gcc-arm-linux-gnueabi
sudo apt-get install g++-arm-linux-gnueabi
```

下载最新的的交叉编译工具，我们组使用的版本

```
gcc version 5.4.0 20160609 (Ubuntu/Linaro 5.4.0-6ubuntu1~16.04.4)
```

修改 qtbase/mkspecs/linux-arm-gnueabi-g++/qmake.conf

增加以下两行

```
QMAKE_CFLAGS_RELEASE += -march=armv7-a -mcpu=cortex-a9
QMAKE_CXXFLAGS_RELEASE + -march=armv7-a -mcpu=cortex-a9
```

意义为 armv7-a 架构，cortex-a9 的 CPU

修改 qt.pro 文件，注释掉 83-84 行

```
# addModule(qt3d, qtdeclarative qtimageformats)
# addModule(qtcanvas3d, qtdeclarative)
```

因为 qt3d 需要用到 openGL，不取消的话，编译会报错

使用以下命令配置

$ ./configure -prefix ../qt5 -opensource -release -confirm-license -xplatform linux-arm-gnueabi-g++ -no-gif -no-nis -no-opengl -no-cups -no-glib -no-dbus -no-rpath -no-openssl -nomake tools -nomake examples -no-sql-sqlite -no-iconv -v

$ make

$ make install

拷贝进根文件系统，烧写。运行代码，发现错误

无法找到 GLIBC2.11 和 GLIBC2.15

原因：编译时使用非常新的交叉编译工具，导致使用的 GLIBC 版本比根文件系统高。

解决方法：将编译器的的库文件拷贝到根文件系统中，得以解决。

修改过后，烧写运行，终端显报错

$ defaultServiceProvider::requestService(): no service found for -"org.qt-project.qt.camera"

找不到 camera 的服务，查找资料，检查项目的 pro 文件，发现 QT5 不再支持 v4l, 使用 Linuxfb 的情况下，需要移植 gstreamer 才能提供 camera 服务。

然后尝试使用 EGLFS 的 platform，从编译好的安卓代码中拷贝 opengl 的头文件和库文件,重新配置，编译。完成后，发现依然没有 camera 插件，仍需移植 gstreamer。

多方尝试均无果，浪费了大量时间，最终决定使用 QT4+v4l2.

QT4 移植：

按照《基于 Cortex-A9 ANDROID&LINUX 系统与应用开发（s4418）》的教程移植。

修改 mkspecs/qws/linux-arm-g++/qmake.conf

修改前：

```
10 # modifications to g++.conf
11 QMAKE_CC                = arm-linux-gcc
12 QMAKE_CXX               = arm-linux-g++
13 QMAKE_LINK              = arm-linux-g++
14 QMAKE_LINK_SHLIB        = arm-linux-g++
15
16 # modifications to linux.conf
17 QMAKE_AR                = arm-linux-ar cqs
18 QMAKE_OBJCOPY           = arm-linux-objcopy
19 QMAKE_STRIP             = arm-linux-strip
20
```

修改后：

```
10 # modifications to g++.conf
11 QMAKE_CC                = arm-none-linux-gnueabi-gcc
12 QMAKE_CXX               = arm-none-linux-gnueabi-g++
13 QMAKE_LINK              = arm-none-linux-gnueabi-g++
14 QMAKE_LINK_SHLIB        = arm-none-linux-gnueabi-g++
15
16 # modifications to linux.conf
17 QMAKE_AR                = arm-none-linux-gnueabi-ar cqs
18 QMAKE_OBJCOPY           = arm-none-linux-gnueabi-objcopy
19 QMAKE_STRIP             = arm-none-linux-gnueabi-strip
```

配置和编译：

$ echo "yes" | ./configure -prefix ../qt4.8.3 -opensource -embedded arm -xplatform qws/linux-arm-g++ -qt-libmng -qt-libjpeg -qt-mouse-pc -no-mouse-linuxtp -no-qt3support -little-endian

$ make

$ make install

QT 编译完成。将../qt4.8.3 中的 lib 和 plugins 拷贝到制作好根文件系统中。