

《嵌入式系统》

（第三次实验 简化版）

厦门大学信息学院软件工程系 曾文华

2023年10月31日

目录

一、电子钟

二、小键盘控制的电子钟

三、摄像头采集实验

四、音视频播放实验

五、红外对射传感器实验

六、蜂鸣器实验

七、交通灯实验

八、**EEPROM**实验

九、陀螺仪实验

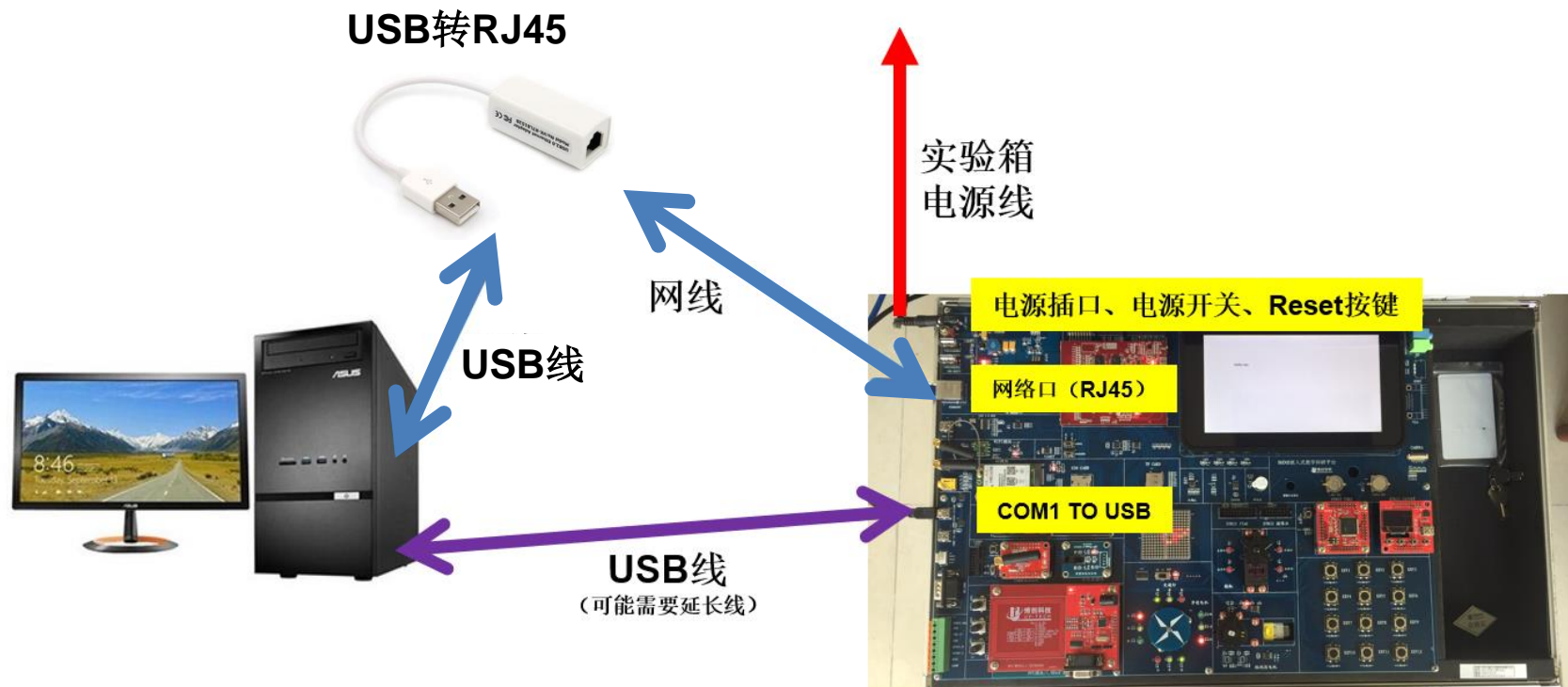
十、**NFC**模块实验

十一、**NFC**应用：一卡通食堂**POS**机

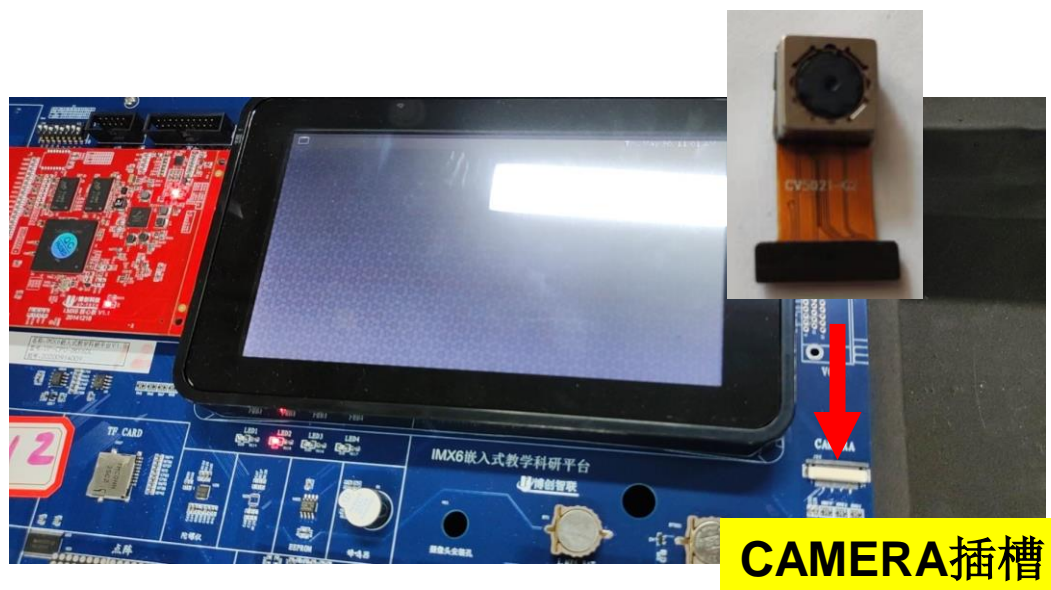
十二、**Qt Creator**环境下的综合实验

第一部分：接线、挂载Ubuntu到 实验箱

接线



微型摄像头插到CAMERA插槽中



打开Ubuntu， 设置IP地址

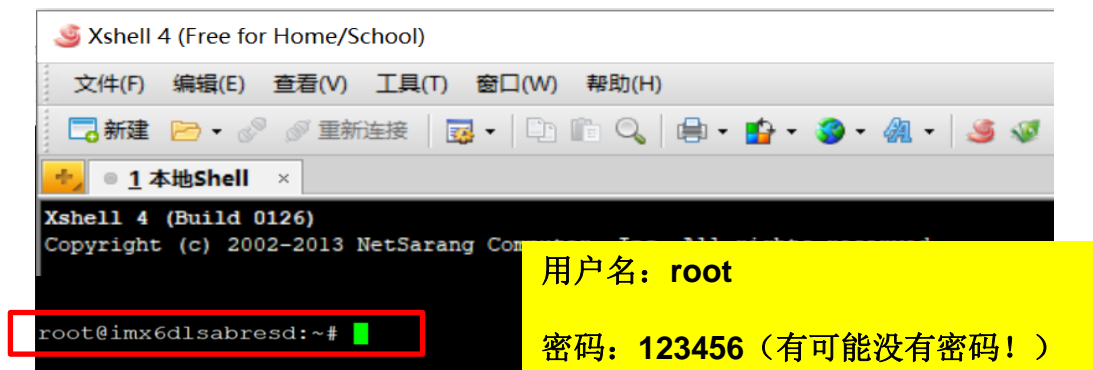
- 打开Ubuntu的“终端”， 执行：

- `sudo su` （密码123456）

- `ifconfig ens33 192.168.33.129`

运行“Xshell 4”，连接实验箱

- 出现下面的界面，表示连接成功！



- 如果出现下面界面，表示连接失败。请按实验箱“复位”键，重启实验箱的Linux系统。

```
Type 'help' to learn how to use Xshell prompt.  
Xshell:\> Xshell:\> Xshell:\>
```

设置实验箱的IP地址，并将 Ubuntu挂载到实验箱

- 在“超级终端（Xshell 4）”上，输入以下命令设置实验箱的IP地址：
 - `ifconfig eth0 192.168.33.155`
 - `mount -t nfs 192.168.33.129:/imx6 /mnt`
- 如果挂载不成功，先在“超级终端（Xshell 4）”执行以下命令查看是否Ping通Ubuntu：
 - `ping 192.168.33.129`
- 然后再Ubuntu的“终端”上执行以下命令，查看是否Ping通实验箱：
 - `ping 192.168.33.155`

第二部分：运行实验程序

1、电子钟

执行电子钟程序

- 在实验箱的“超级终端（Xshell 4）”上执行：
 - `cd /mnt/whzeng/clock`
 - `./clock 24:59:50`
 - 则显示“input error!”，并退出程序：
 - `./clock 23:60:50`
 - 则显示“input error!”，并退出程序：
 - `./clock 23:59:60`
 - 则显示“input error!”，并退出程序：
 - `./clock 23:59:50`
 - 则数码管上显示：23-59-50，并开始变化时间



在Ubuntu上执行电子钟程序

- 在Ubuntu的“终端”上，执行：

- `cd /imx6/whzeng/clock_pc`
 - `./clock_pc 23:59:50`

按Ctrl+C，退出程序

- Ubuntu的“终端”上显示：

```
uptech@uptech:~$ cd /imx6/whzeng/clock_pc
uptech@uptech:/imx6/whzeng/clock_pc$ ./clock_pc 23:59:50

2 3 - 5 9 - 5 1
```

- 退出程序后，需要关闭Ubuntu的“终端”，再重新打开

2、小键盘控制的电子钟

执行小键盘控制的电子钟程序

- 在实验箱的“超级终端（Xshell 4）”上执行：
 - `cd /mnt/whzeng/key_clock`
 - `./key_clock`
- 按小键盘的数字键（**2 3 5 9 5 0**），设置时钟的初值；如果设置错误（如果：**2 4 5 9 5 0**），则按小键盘的“**Clean**”键，清除，然后重新设置
- 按小键盘的“**Start**”键，启动时钟，数码管显示变化的时间；按小键盘的“**Stop**”键，停止时钟，数码管的时间停止变化



在Ubuntu上执行的小键盘控制的电子钟程序

- 在Ubuntu的“终端”上，执行：

- `cd /imx6/whzeng/key_clock_pc`
- `./key_clock_pc`

```
uptech@uptech:/imx6/whzeng/key_clock_pc$ ./key_clock_pc  
2 3 - 5 9 - 5 0
```

- 通过电脑键盘输入：**2 3 5 9 5 0** （“-”会自动显示）
- 如果输入的初值不正确（**2 4 5 9 5 0**），则按电脑的“**t**”键，然后重新输入时钟初值
- 通过电脑键盘输入：**s** 则时钟开始计时
- 通过电脑键盘输入：**t** 则时钟停止计时

3、摄像头采集实验

(1) 命令方式的摄像头采集实验

(2) 摄像头采集程序 (**camera.c**)

(3) **Qt Creator**程序摄像头采集实验 (**qt-camera**)

(1) 运行摄像头采集命令

- 在实验箱的“超级终端（Xshell 4）”上执行：

- `cd /unit_tests/V4L2`

- `./mxc_v4l2_overlay.out -iw 640 -ih 480 -it 0 -il 0 -ow 400 -oh 400 -ot 20 -ol 20 -r4 -t 50 -d 0 -fr 30`

30修改为50，则增加采集时间长度

`mxc_v4l2_overlay.out` 是摄像头采集命令

`-iw 640 -ih 480 -it 0 -il 0 -ow 400 -oh 400 -ot 20 -ol 20 -r4 -t 50 -d 0 -fr 30` 为参数

- 此时，实验箱的液晶屏上将显示拍摄的视频，用手放到摄像头的上面，观看是否拍摄到
 - 如果摄像头不工作，请按Reset键，重启实验箱



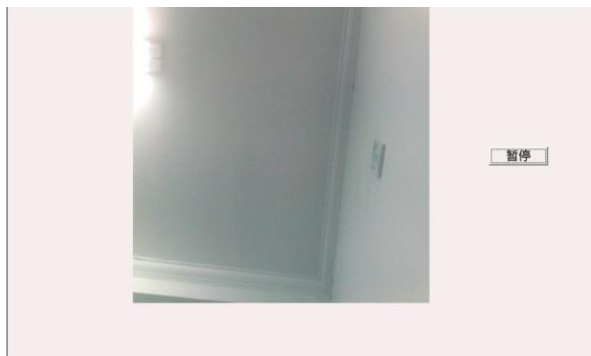
(2) 运行摄像头采集程序

- 在实验箱的“超级终端（Xshell 4）”上执行：
 - `cd /mnt/whzeng/camera`
 - `./camera`
- 此时，实验箱的液晶屏上将显示拍摄的视频，用手放到摄像头的上面，观看是否拍摄到



（3）运行Qt摄像头采集程序

- 方式一：在实验箱的“超级终端（Xshell 4）”上执行：
 - `cd /mnt/wzheng/qt-camera`
 - `./camera`
 - 此时，实验箱的液晶屏上会显示摄像头采集到的视频
 - 按液晶屏上的“播放”按钮则播放采集到的视频，按“暂停”按钮会停止视频的采集



- 方式二：在Ubuntu的“终端”上执行：

- `sudo sh /opt/qtcreator-4.5.0/bin/qtcreator.sh`

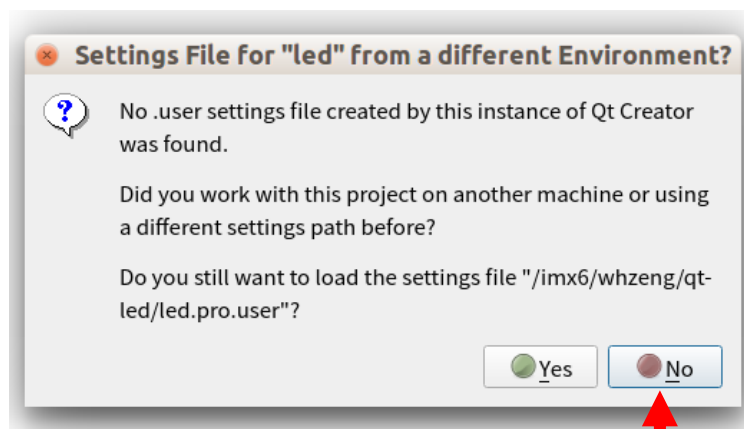
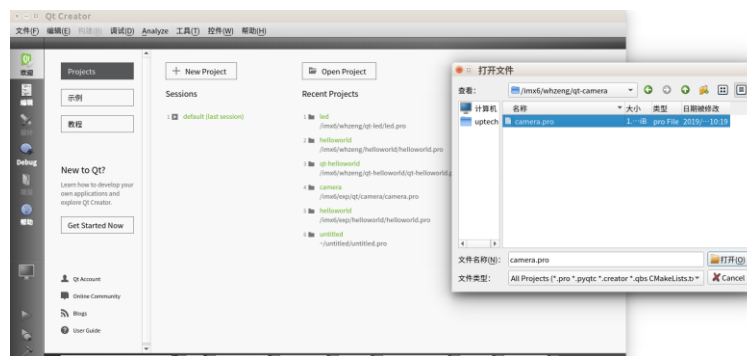


- 测试Qt Creator是否与实验箱连接成功

- 点击Qt Creator的菜单“工具”->“选项”->“设备”。
 - 如果“主机名称”中的IP地址不是实验箱的IP地址，请修改成实验箱的IP地址：**192.168.33.155**。
 - 点击“Test”，则连接实验箱，如果连接成功，则显示左下角的界面。
 - 如果连接不成功，则显示右下角的界面，此时请检查虚拟机（Ubuntu）和实验箱之间是不是能够Ping通过？



- 打开qt-camera工程

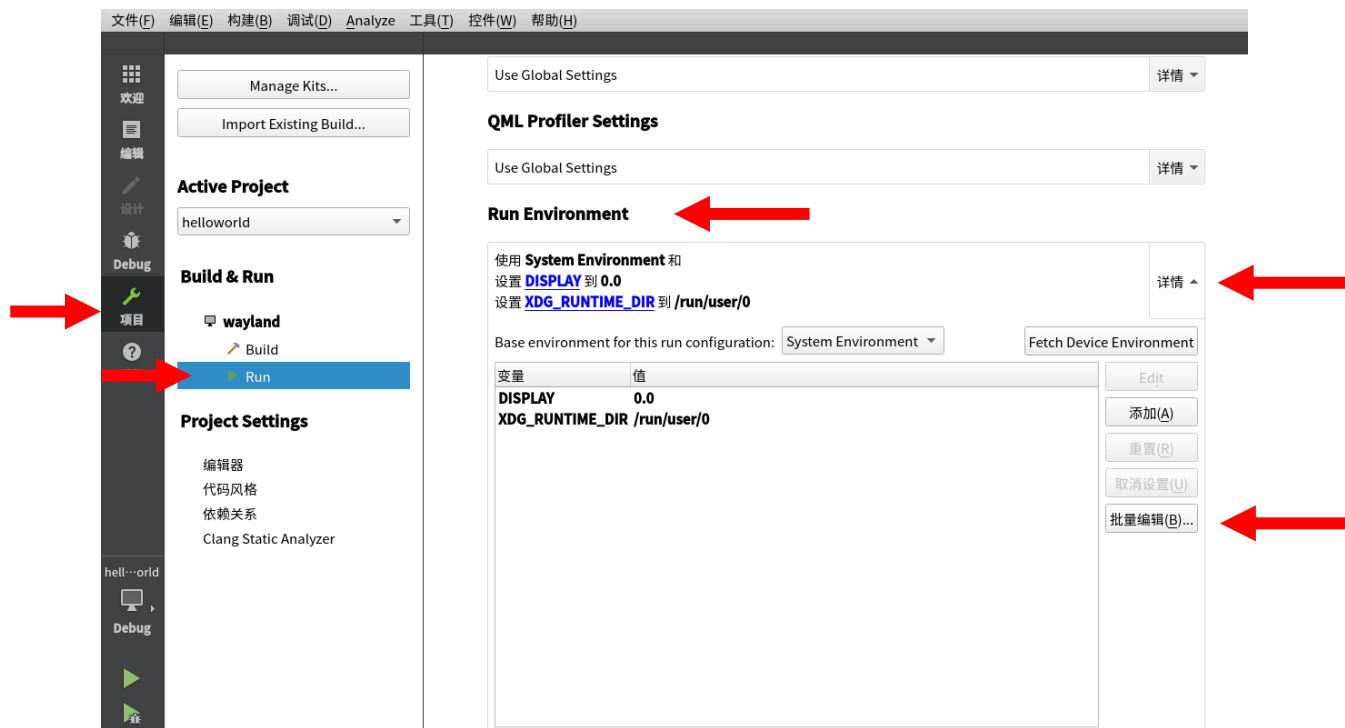


这里选 “No”

- 查看“运行环境”变量

- 点击Qt Creator的“项目”->“Run”，将界面拉到最下面的“Run Environment”

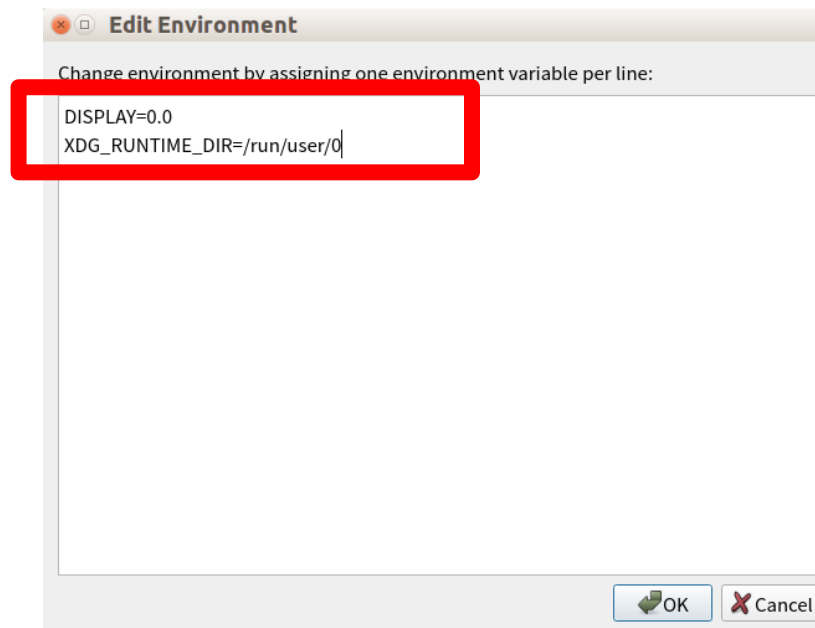
- 点击“详情”、“批量编辑”



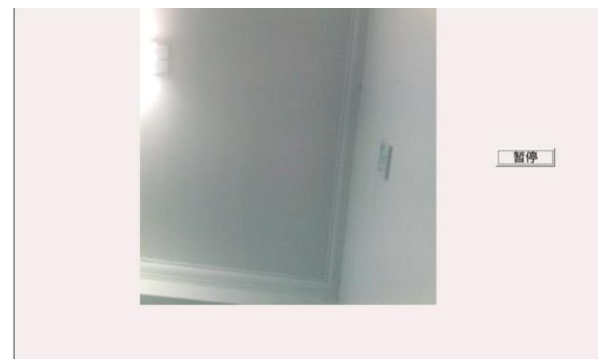
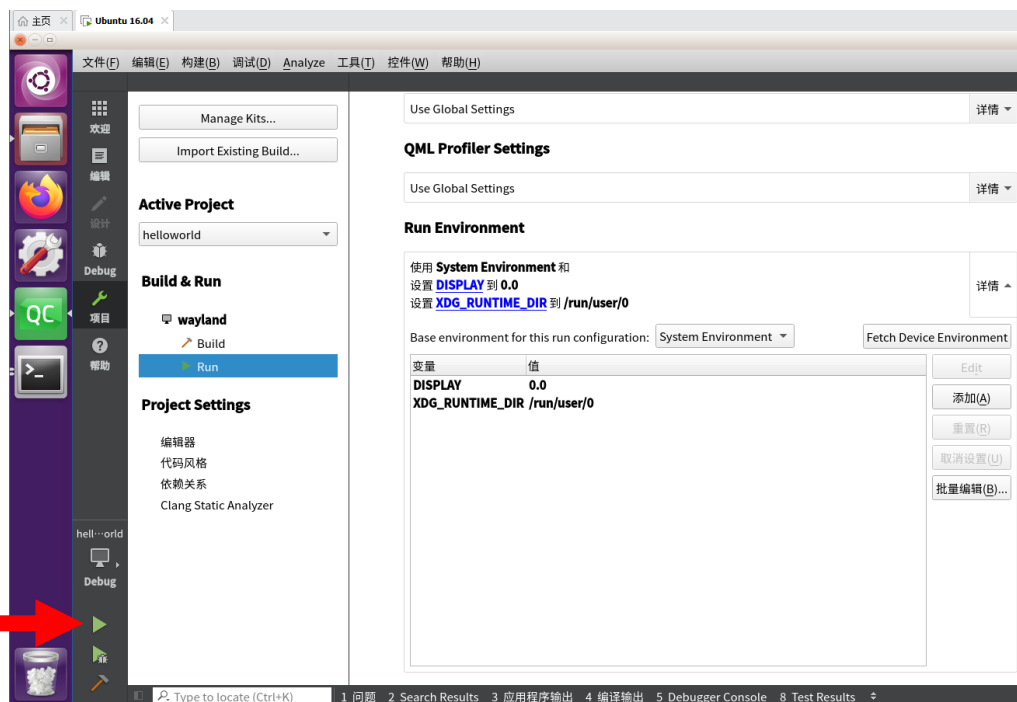
- 查看“运行环境”中是不是如下的内容，如果不是，则增加增加以下内容：

DISPLAY=0.0

XDG_RUNTIME_DIR=/run/user/0



- 点击运行按钮（**绿色三角形**）
- 此时，实验箱的液晶屏上会显示摄像头采集到的视频；按液晶屏上的“播放”按钮则播放采集到的视频，按“暂停”按钮会停止视频的采集



4、音视频播放实验

运行音视频播放程序

- 将耳机和麦克风分别插入到实验箱的耳机与麦克风插孔中
- (1) 播放视频。在实验箱的“超级终端 (Xshell 4)” 上执行：

- `cd /mnt/whzeng/audio_video`

- `gplay-1.0 12dao_feng_wei_flv.avi`

`gplay-1.0` 是播放视频的命令

`12dao_feng_wei_flv.avi` 为视频文件（同学们可以更换为其他视频文件）

- 此时实验箱的液晶屏将显示播放的视频，需要通过耳机听声音



耳麦插孔

运行音视频播放程序

- (2) 播放音频。在实验箱的“超级终端 (Xshell 4)”上执行：

- **aplay 1.wav**

- 此时在耳机中会听到播放的声音

aplay 是播放音频的命令

1.wav 为音频文件（同学们可以更换为其他音频文件）

- (3) 设置音量（声音最大为**127**，最小为**0**）。在实验箱的“超级终端 (Xshell 4)”上执行：

- **amixer set Headphone 127**

- **aplay 1.wav**

- 此时，耳机中的声音很大

- **amixer set Headphone 90**

- **aplay 1.wav**

- 此时，耳机中的声音很小

运行音视频播放程序

- (4) 录音。在实验箱的“超级终端 (Xshell 4)”上执行：

- `amixer sset 'MIXINR IN3R' on`
- `amixer sset 'INPGAR IN3R' on`
- `amixer cset name='Capture Volume' 63,63`
- `arecord -r 44100 -f S16_LE -c 2 -d 10 123.wav`
- 10为录用时间长度，可以调整
- 此时对着麦克风讲话，讲话的声音将被录音

`amixer`命令设置相关的参数

`arecord` 是录音的命令

`-r 44100 -f S16_LE -c 2 -d 10` 为相关参数

`123.wav` 为保存的录音文件

- (5) 播放录好的声音。在实验箱的“超级终端 (Xshell 4)”上执行：

- `aplay 123.wav`
- 此时在耳机中会听到录好的声音

5、红外对射传感器实验

运行红外对射传感器实验

- 在实验箱的“超级终端（Xshell 4）”上执行：

- `cd /mnt/whzeng/irda`

- `./irda`

- 此时，用纸张（要厚一点的）挡住红外模块的槽口，电脑上显示：

- `irda interrupt!`

- 并显示读取的串口内容



按**Ctrl+C**退出程序

6、蜂鸣器实验

- (1) 基本的蜂鸣器实验 (**beep.c**)
- (2) 只发出一声响的蜂鸣器实验 (**beep-1.c**)
- (3) 发出5声长短响的蜂鸣器实验 (**beep-2.c**)

运行蜂鸣器程序

- 在实验箱的“超级终端（Xshell 4）”上执行：

- `cd /mnt/whzeng/beep`
- `./beep`

按**Ctrl+C**退出程序

- 此时，蜂鸣器会发出滴滴的声音

- 在实验箱的“超级终端（Xshell 4）”上执行：

- `cd /mnt/whzeng/beep-1`
- `./beep-1`

- 此时，蜂鸣器会只发出一声响

- 在实验箱的“超级终端（Xshell 4）”上执行：

- `cd /mnt/whzeng/beep-2`
- `./beep-2`

- 此时，蜂鸣器会发出5声长短响



7、交通灯实验

运行交通灯程序

- 在实验箱的“超级终端（Xshell 4）”上执行：

- `cd /mnt/whzeng/light`

- `./light`

- 此时，12个LED灯将模拟十字路口的交通灯状态。



8、EEPROM实验

运行EEPROM程序

- 在实验箱的“超级终端（Xshell 4）”上执行：
 - `cd /mnt/whzeng/eeprom`
 - `./eeprom`

— 电脑上将显示：

- `writeok`
- `readok`
- `buff = 12345678`
- `writeok`
- `readok`
- `buff = abcdefgh`

9、陀螺仪实验

(1) 基本陀螺仪实验 (`gyroscope.c`)

(2) 高级陀螺仪实验 (`gyroscope-1.c`)

更新程序并重新编译

- 将新的陀螺仪实验程序（**gyroscope.c**）拷贝到Ubuntu的 /imx6/whzeng/gyroscope 下，替换掉原来的文件
- 将新的陀螺仪实验程序（**gyroscope-1.c**）拷贝到Ubuntu的 /imx6/whzeng/gyroscope-1 下，替换掉原来的文件
- 在Ubuntu的“终端”上编译**gyroscope.c**程序
 - `cd /imx6/whzeng/gyroscope`
 - `source /opt/fsl-imx-wayland/4.9.88-2.0.0/environment-setup-cortexa9hf-neon-poky-linux-gnueabi`
 - `make clean`
 - `make`
- 在Ubuntu的“终端”上编译**gyroscope-1.c**程序
 - `cd /imx6/whzeng/gyroscope-1`
 - `source /opt/fsl-imx-wayland/4.9.88-2.0.0/environment-setup-cortexa9hf-neon-poky-linux-gnueabi`
 - `make clean`
 - `make`

运行陀螺仪程序1

- 在实验箱的“超级终端（Xshell 4）”上执行：

- `cd /mnt/whzeng/gyroscope`

- `./gyroscope /dev/input/event2 1`

`/dev/input/event2`表示陀螺仪的设备号

`1`表示启动陀螺仪

- 电脑上将显示X、Y、Z三个方向的坐标值
- 此时，**搬起实验箱，摇动实验箱**，X、Y、Z坐标值会发生变化

运行陀螺仪程序2

- 运行另一个陀螺仪应用程序，在实验箱的“超级终端（Xshell 4）”执行：

- `cd /mnt/whzeng/gyroscope-1`

- `./gyroscope-1 /dev/input/event2 1`

`/dev/input/event2`表示陀螺仪的设备号

`1`表示启动陀螺仪

- 电脑上将显示X、Y、Z三个方向的加速度、角度、角速度

- 此时，**搬起实验箱，摇动实验箱**，X、Y、Z加速度、角度、角速度会发生变化

10、NFC模块实验

(1) 基本的NFC模块实验 (nfc.c)

(2) 在LED点阵上显示卡号的NFC模块实验 (nfc-1.c)

运行NFC程序

- 在实验箱的“超级终端（Xshell 4）”上执行：

- `cd /mnt/whzeng/nfc`
 - `./nfc`

按Ctrl+C退出程序

- 此时将实验室提供的一张NFC卡，以及你的校园卡，分别放到NFC模块上，则得到不同的uid值：
`uid=0xxxxxxxxx`

- 将上述两张卡的uid值拷贝下来，然后复制到nfc-1.c文件中的相关位置（见下一页）

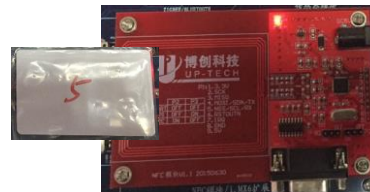
- 在Ubuntu的“终端”上编译nfc-1.c程序

- `cd /imx6/whzeng/nfc-1`
 - `source /opt/fsl-imx-wayland/4.9.88-2.0.0/environment-setup-cortexa9hf-neon-poky-linux-gnueabi`
 - `make clean`
 - `make`

- 在实验箱的“超级终端（Xshell 4）”上执行：

- `cd /mnt/whzeng/nfc-1`
 - `./nfc-1`

- 此时再将实验室提供的一张NFC卡，以及你的校园卡，分别放到NFC模块上，会看到LED点阵上分别显示“1”“2”，在电脑上显示“Card ID = x”



卡的UID代码转换为卡号函数

nfc-1.c程序

```
int card_number(unsigned long uid)
{
    switch(uid)
    {
        case 0x526d13d5:
            return 1;

        case 0x82c70cd5:
            return 2;

        case 0xc579860f:
            return 3;

        case 0xb26426d5:
            return 4;

        case 0xf516900f:
            return 5;

        case 0xf2171fd5:
            return 6;

        case 0x453e890f:
            return 7;

        case 0x153f8c0f:
            return 8;

        case 0xcb273de6:
            return 9;

        default:
            return 0;
    }
}
```

用实验室的NFC卡的ID替换这里的ID值

用你的校园卡的ID替换这里的ID值

11、NFC应用：一卡通食堂POS机

修改并编译程序

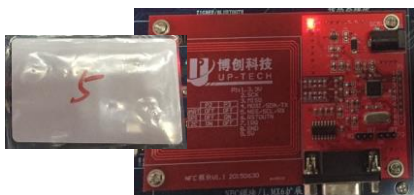
- 将前面实验的nfc-1.c中的2个ID值拷贝到nfc-pos.c程序中相应的位置:

```
int card_number(unsigned long uid)
{
    switch(uid)
    {
        case 0x526d13d5:
            return 1;

        case 0x82c70cd5:
            return 2;
```

nfc-pos.c程序

- 在Ubuntu的“终端”上执行:
 - `cd /imx6/whzeng/nfc-pos`
 - `source /opt/fsl-imx-wayland/4.9.88-2.0.0/environment-setup-cortexa9hf-neon-poky-linux-gnueabi`
 - `make clean`
 - `make`



运行程序



- 在实验箱的“超级终端（Xshell 4）”上执行：
 - `cd /mnt/whzeng/nfc-pos`
 - `./nfc-pos`
 - Do you want to initialize the card_money.txt & card_record.txt? (y or n)
 - 输入“**Y**”，则初始化card_money.txt文件（卡中余额）和card_record.txt文件（消费记录），初始化后，卡中余额和消费记录全部清零
 - 输入“n”，则不初始化card_money.txt文件和card_record.txt文件
- 向卡中充值**：通过小键盘的数字键输入充值金额（例如：**100**），按小键盘的“*”键，可以看到数码管显示**-100**；再按小键盘的“*”键，则显示**P1 00**。将NFC卡放到读卡器上，则向卡中充值。如果输入金额错误，则按小键盘的“#”键，取消，然后再次输入。将卡从读卡器上拿开。
- 消费**：通过小键盘的数字键输入消费金额（例如**15**），按小键盘的“*”键，可以看到数码管显示**-15**。将NFC卡放到读卡器上，则从卡中充扣除消费金额，数码管上显示扣除后的余额：**85**。如果输入金额错误，则按小键盘的“#”键，取消，然后再次输入。将卡从读卡器上拿开。
- 再次消费**：通过小键盘的数字键输入消费金额（例如**18**），按小键盘的“*”键，可以看到数码管显示**-18**。将NFC卡放到读卡器上，则从卡中充扣除消费金额，数码管上显示扣除后的余额：**67**。如果输入金额错误，则按小键盘的“#”键，取消，然后再次输入。将卡从读卡器上拿开。
- 查询余额**：将NFC卡放在读卡器上，此时数码管显示卡中余额**67**。将卡从读卡器上拿开。将卡从读卡器上拿开。
- 查询消费记录**：按小键盘的“*”键（上翻键）和“#”键（下翻键），可以查询消费记录。分别是**15**和**18**。
- 操作过程中，**LED点阵**会显示NFC卡的卡号，**蜂鸣器**会发出相应的声音。



12、Qt环境下的综合实验

运行Qt环境综合程序（方式一）

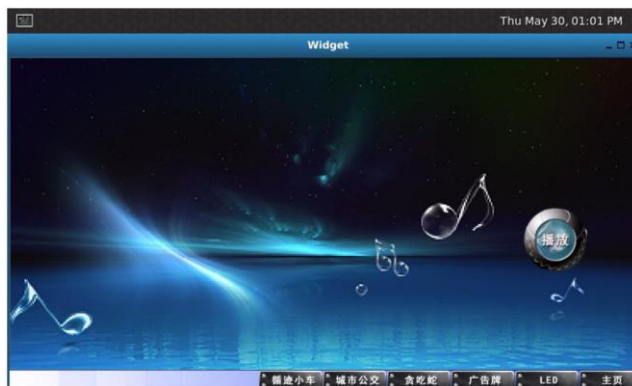
- 在实验箱的“超级终端（Xshell 4）”上执行：

- `cd /mnt/wzheng/imx6_V1_0`

- `./imx6_V1_0`

- 此时，实验箱的液晶屏上会显示综合实验的界面

- 请根据界面的内容进行相关的操作



运行Qt环境综合程序（方式二）

- 在Ubuntu的“终端”上执行：

— `sudo sh /opt/qtcreator-4.5.0/bin/qtcreator.sh`



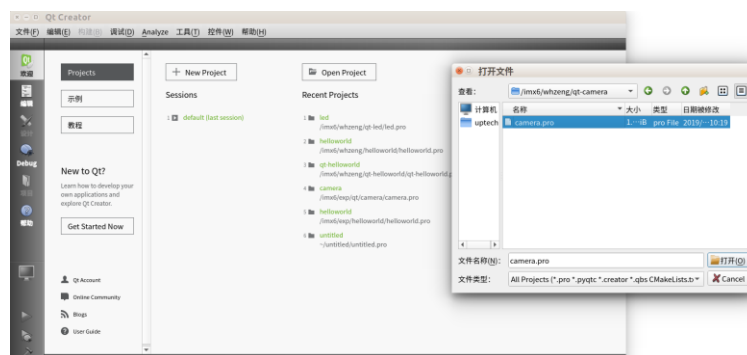
- 测试Qt Creator是否与实验箱连接成功

- 点击Qt Creator的菜单“工具”->“选项”->“设备”。
- 如果“主机名称”中的IP地址不是实验箱的IP地址，请修改成实验箱的IP地址：**192.168.33.155**。
- 点击“Test”，则连接实验箱，如果连接成功，则显示左下角的界面。
- 如果连接不成功，则显示右下角的界面，此时请检查虚拟机（Ubuntu）和实验箱之间是不是能够Ping通过？



运行Qt环境综合程序（方式二）

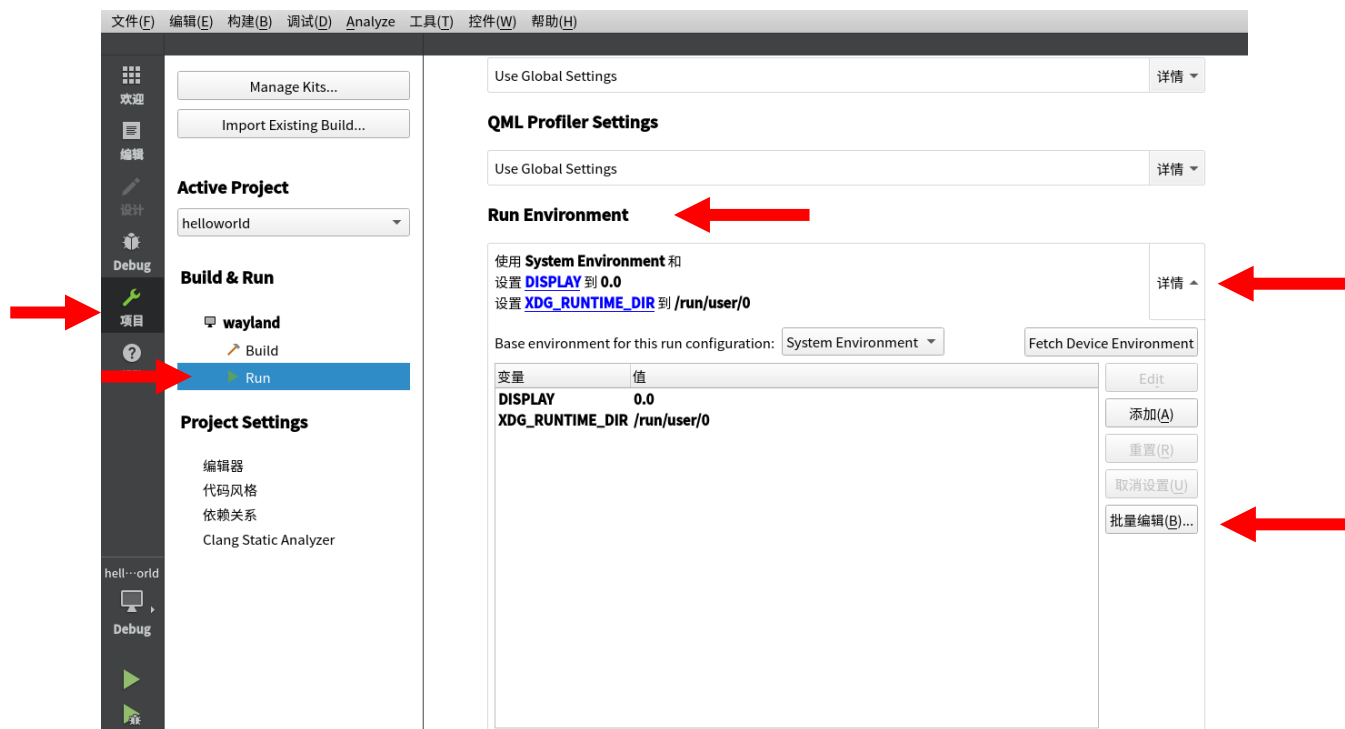
- 打开imx6_V1_0工程



这里选 “No”

运行Qt环境综合程序（方式二）

- 查看“运行环境”变量
 - 点击Qt Creator的“项目”->“Run”，将界面拉到最下面的“Run Environment”
 - 点击“详情”、“批量编辑”

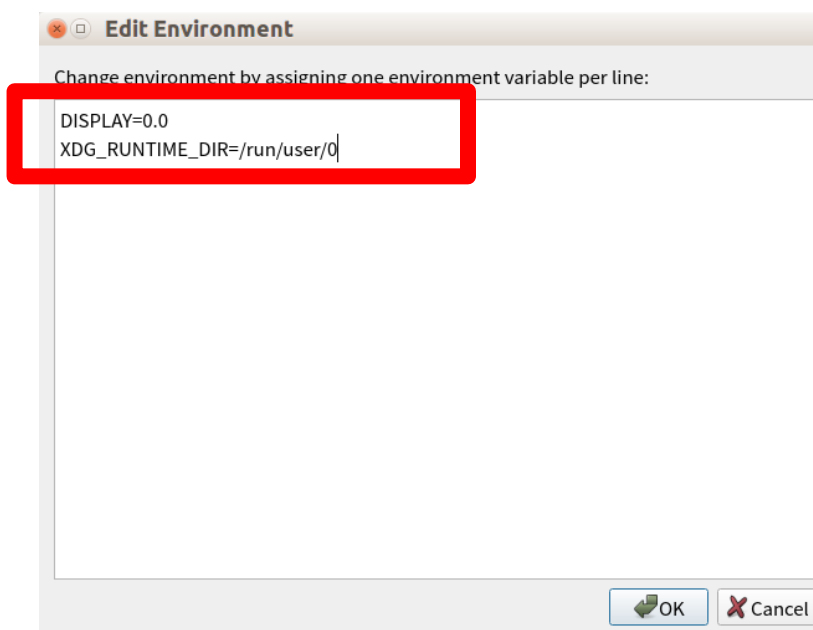


运行Qt环境综合程序（方式二）

- 查看“运行环境”中是不是如下的内容，如果不是，则增加增加以下内容：

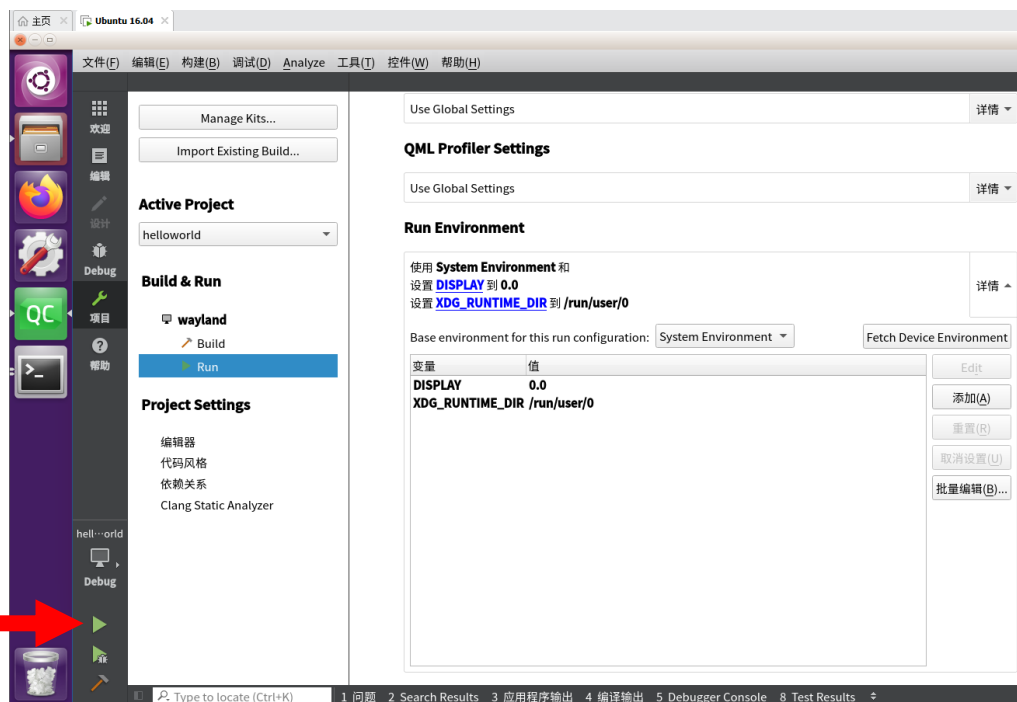
DISPLAY=0.0

XDG_RUNTIME_DIR=/run/user/0



运行Qt环境综合程序（方式二）

- 点击运行按钮（**绿色三角形**）
- 此时，实验箱的液晶屏上会显示摄像头采集到的视频；按液晶屏上的“播放”按钮则播放采集到的视频，按“暂停”按钮会停止视频的采集



Thanks