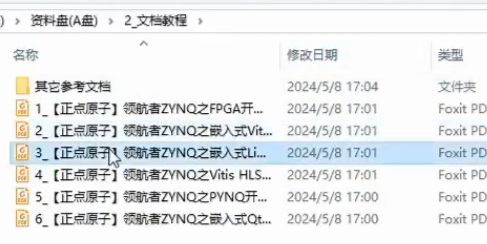


主要控制的外设有：



参考文档还是 开发指南：



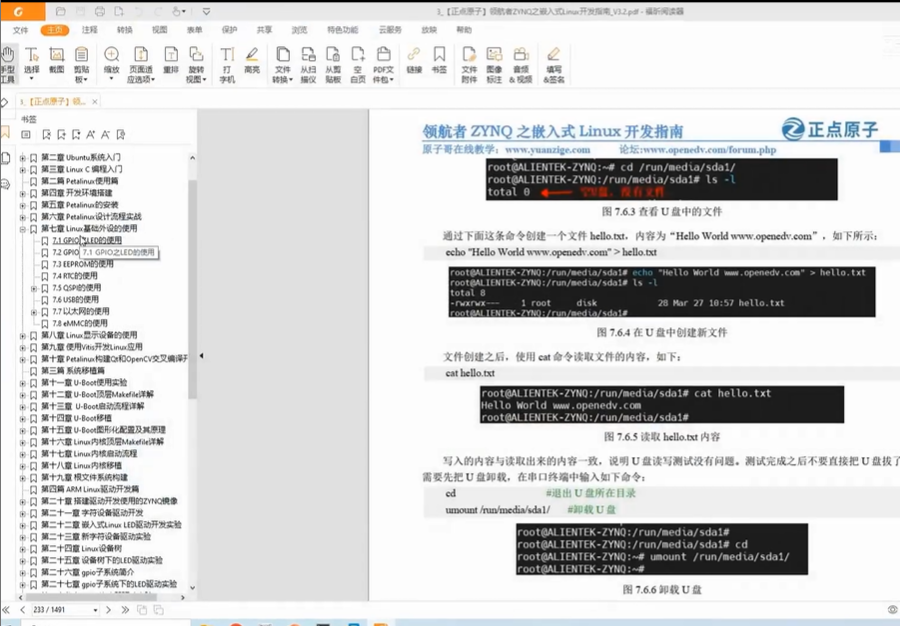
LED和按键 都是GIPO控制的。

EEPROM和RTC是挂载到IIC下面的。

然后是QSPI、USB、以太网、eMMC

打开串口调试助手MobaXterm,上电开发板，等待启动。

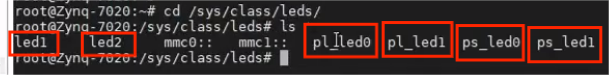
由于开发板也是Linux的Ubuntu系统。



在之前介绍过：Linux下一切皆文件。我们首先来看看GPIO是如何控制的：

首先，执行命令：cd /sys/class/leds

然后 ls 查看下面的设备：



这几个对应6个LED灯。

执行命令cd led1进入LED1目录下面。然后有如下选项：

IMG_256

先获取brightness(亮度)的内容：cat brightness:

IMG_256

再获取最大亮度：max\_brightness:

IMG_256

此时我们向brightness输入值：

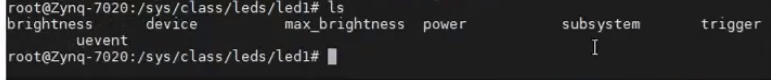
echo 255 > brightness

echo 0 > brightness

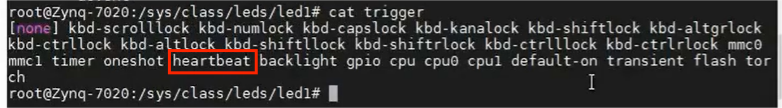
从而实现对LED1的亮灭控制。



可以看到除了brightness还有一个trigger（触发）。



执行cat trigger 可以看到：



首先我们将led1点亮：

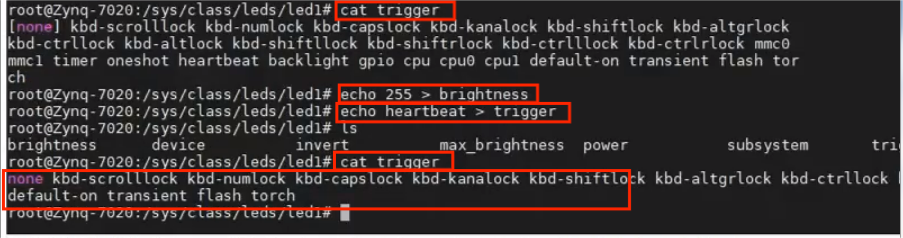
echo 255 > brightness

再将触发方式trigger改成：heartbeat（心跳），他就会不停闪烁。

echo heartbeat > trigger

（通过对trigger文件的操作，实现对触发方式的控制。）

而这些触发方式，在内核中都是写好的，我们直接echo一个值来修改即可。



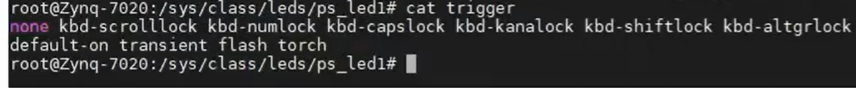
由于刚刚echo 了heartbeat，所以heartbeat就被选中了。

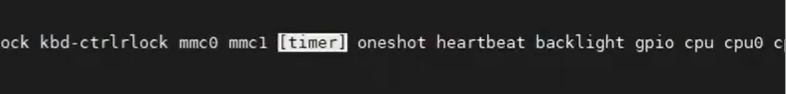
如果echo none就没有触发方式。

执行cd ps\_led1进入ps\_led1

ls可以看到;

IMG_256

cat 获取ps\_led1的trigger的内容:查看他的触发方式。  




可以看到是timer，所以Led灯是闪烁的。

上图default-on模式是默认打开，也就是常亮。

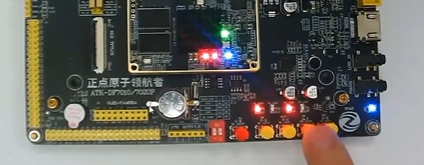
LED就介绍完了：这里主要介绍两个地方：

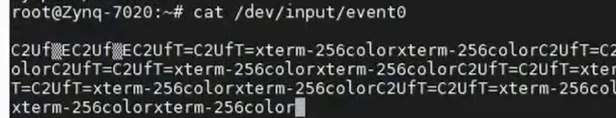
1. 通过这个brightness这个文件来控制亮度。向里面输入0~255调整亮度
2. trigger文件，通过向里面写入不同的值来控制触发方式。

2.按键

输入命令：cat /dev/input/event0

每按一次按键，对应地就会输出一些数据





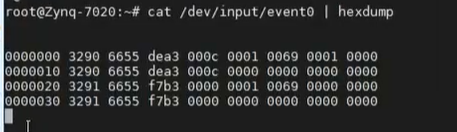
可以使用ctrl + c退出接收按键信息。

我们可以看到输出信息是一些乱码，为了调整格式：

输入命令：cat /dev/input/event0 | hexdump

(管道命令)

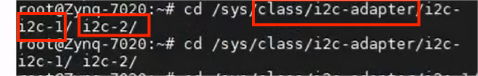
此时输出为：



3.EEPROM:

我们知道这是一个存储设备：

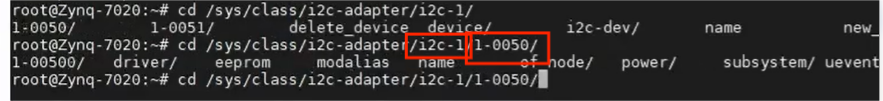
我们需要在class下面的IIC适配器（i2c-adapter）里面去找：



我们可以看到IIC适配器（i2c-adapter）里面有两个文件：

I2c-1和i2c-2

我们进入I2c-1。里面有一个1-0050：



1-0050里面就有eeprom。

IMG_256

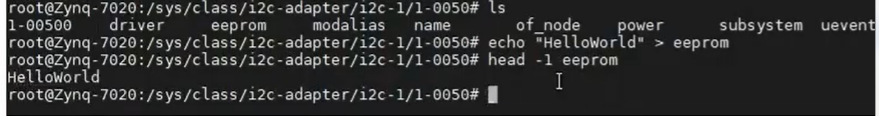
我们如何知道它的具体位置？

在设备树里面可以清楚地看到。

我们上一讲拷贝了设备树，那里面就有对设备树详细的描述。设备树后续再详细介绍。

我们执行 echo “HelloWorls” > eeprom 将字符串输入到eeprom 中

再执行head -1 eeprom命令。将eeprom 的第1行打印出来。



-1表示打印第一行， -10就表示打印前十行。

4.RTC:

我们知道开发板上有两个时钟：

一个是Linux的时钟，另一个是硬件时钟。

硬件时钟就是开发板断电之后，硬件时钟会继续运行，而开发板上有一块纽扣电池，可以提供电源。来保证RTC的运行。来继续记录硬件的时间。

硬件的时钟如何查看？

执行命令：hwclock （hardware clock）即可查看RTC保存的时间。

IMG_256

执行date 也可以打印时间：这个时间就是Linux系统的时钟。

IMG_256

执行date -S “2024-05-28 0:0:0” 修改时间。

IMG_256

可见时间被修改了。

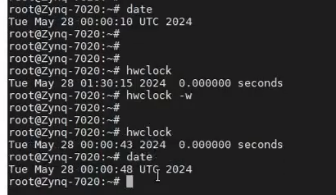
修改后：

使用date命令查看，就是修改后的时间。

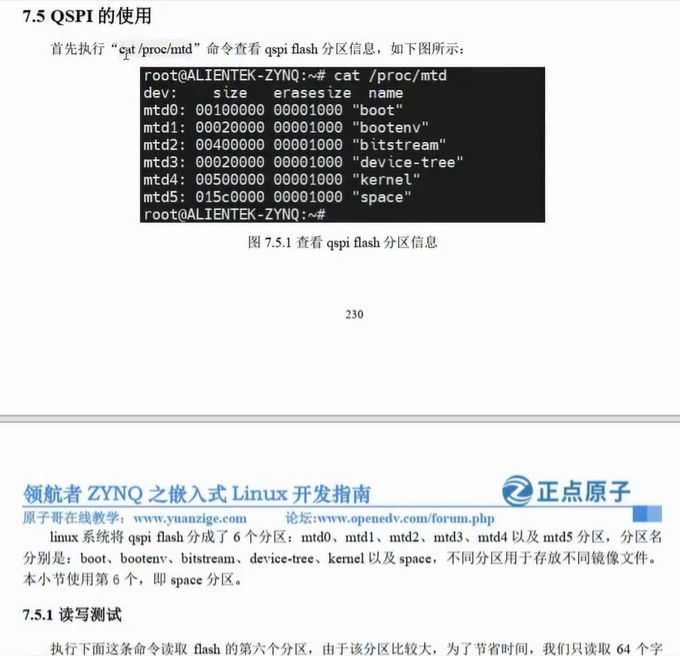
使用hwclock命令查看的，仍然是之前的时间。

那么我们修改完时钟之后，如何放到硬件时钟里面呢？

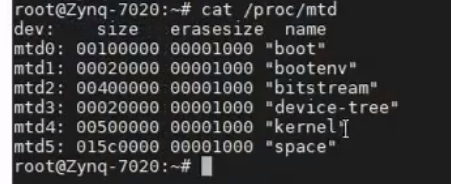
执行：hwclock -w 就可以将系统时钟写入硬件时钟。完成同步。



5.QSPI:



首先执行：cat/proc/mtd 命令查看qspi flash分区信息。



可以看到里面有0~5共6个分区。

分别是boot

bootenv boot的环境变量

比特流

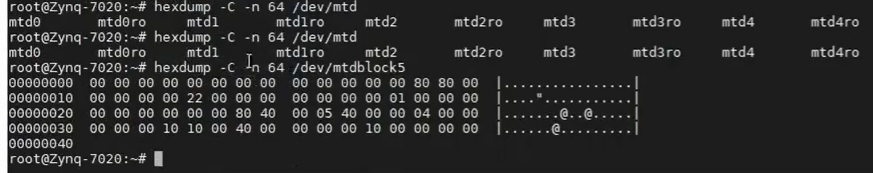
设备树

内核

Space

执行：hexdump -C -n 64 /dev/mtdblock5 命令

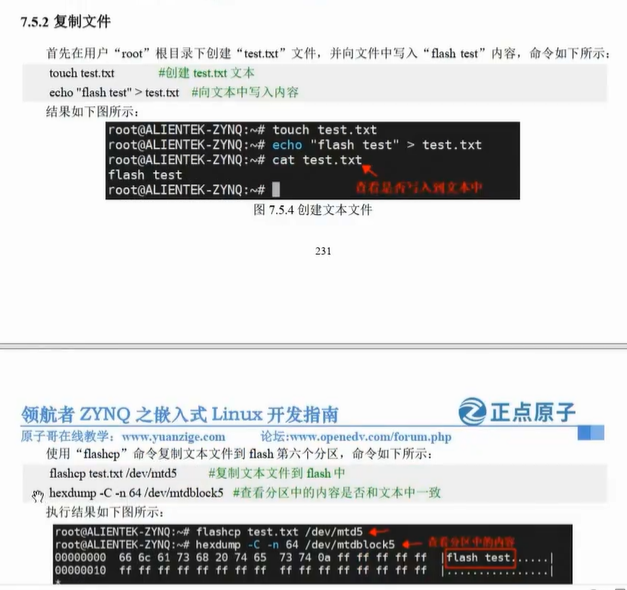
读取flash 第六个分区，长度为64字节。





右边的就是ASSIC码。

左边对应的就是其十六进制文件。



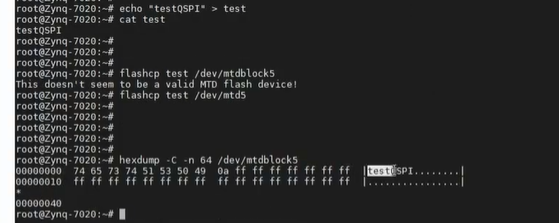
执行touch test,创建一个test文件

执行echo “testQSPI” > test

执行cat test

执行flashcp test /dev/mtd5

执行hexdump -C -n 64 /dev/mtdblock5



此时其打印的就是testQSPI。