嵌入式考试范围：

一、简答题：2道，每题10分

1）makefile，讲出每行代码的意思；

uart.bin: start.o clock.o uart.o main.o

**目标: 依赖**

**下面是命令：（必须以tab开头），命令就是输入make以后需要执行的命令。**

arm-linux-gcc -c -o led.o led.S

**将.S文件编译成.o文件**

arm-linux-ld -Ttext 0xD0020010 -o uart.elf $^

**arm-linux-ld：**链接工具，将多个目标文件链接成一个文件

**-Ttext：**指明elf文件运行的地址为：0xd0020010

**$^：**表示空行

arm-linux-objcopy -O binary uart.elf $@

**arm-linux-objcopy：**复制一个目标文件到另一个中，即进行格式转换

这里将elf文件转换为可执行的二进制文件

**$@：**代表目标文件(target)

arm-linux-objdump -D uart.elf > uart.dis

**arm-linux-objdump -D：**反汇编可执行段，用于显示二进制文件信息

%.o : %.c

arm-linux-gcc -c $< -o $@ -nostdlib

**$<：**代表第一个依赖文件

**-nostdlib：**把指定的文件传递给连接器

%.o:%.S

arm-linux-gcc -c $< -o $@ -nostdlib

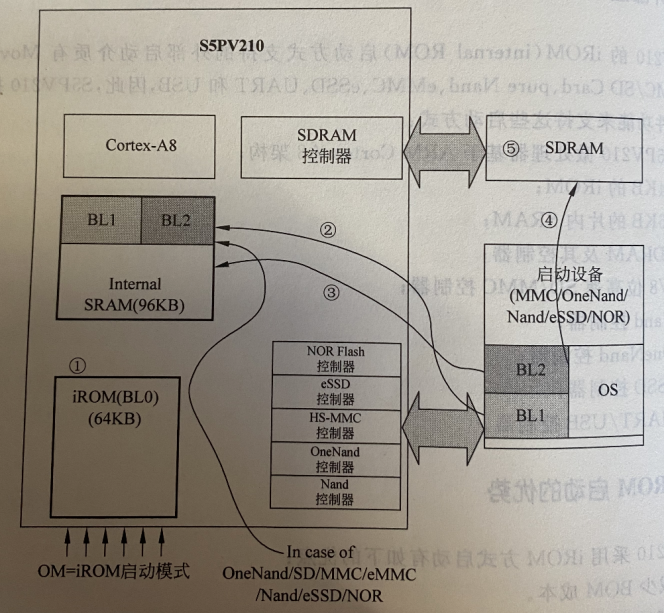
clean:

**清除上次make 产生的文件**

rm \*.o \*.elf \*.bin \*.dis

-c，只执行到编译，输出目标文件

2）S5PV210的启动流程；



1、S5PV210上电后执行iROM中的固化代码，即BL0（Bootloader0），这个代码是厂家出厂前烧写好的，比如进行一些时钟初始化、控制器初始化、启动初始化等；

2、iROM继续执行加载Bootloader到片内SRAM（总大小96KB）中，即BL1（最大不超过16KB），并跳到BL1中执行。

3、执行BL1加载Bootloader剩余部分到SRAM中，即BL2（最大不超过80KB），并跳到BL2中执行。

4、执行BL2初始化DRAM控制器，并加载OS到SDRAM。

5、跳转到OS起始地址处执行。

3）DNW下载的软硬件流程；

1、将实验箱拨码开关拨到2，长按电源键，打开设备管理器，选择对应的驱动，手动添加usb驱动器；

2、关闭电源，回拨开关至off，等待添加的驱动安装成功；

3、打开软件，用串口线连接开发板和电脑，设置开发板拨码开关全为off；

4、打开dnw软件，点击option，设置波特率：115200，串口查看设备管理器后设置为对应口，此处设置为com1，地址设置为0xd0020010；

5、点击connect，迅速连按空格进入uboot；

6、输入dnw 0xd0020010设置下载地址，等待驱动安装完成；

7、点击Transmit，选择编译好的bin文件传输至开发板；

8、输入go 0xd0020010，即可运行文件。

二、填空题：3道，一共35分

初始化+AD编程

GPIO初始化、串口初始化、中断初始化、定时器初始化、定时器中断初始化、AD编程。（系统时钟初始化不考）。

GPIO初始化：

简化表示：GPC0\_3表示[15:12] 书上的初始化是GPC，实验用GPJ

实验：利用按键SW5和SW6控制LED3和LED4。按下按键，相应LED亮，再次按下按键，LED 灭。

初始化：

#define GPJ2CON (\*(volatile unsigned long \*) 0xE0200280)

#define GPJ2DAT (\*(volatile unsigned long \*) 0xE0200284)

#define GPH0CON \*((volatile unsigned long \*) 0xE0200C00)

#define GPH0DAT \*((volatile unsigned long \*) 0xE0200C04)

**int** main()

{

**int** key\_val = 0;

**int** bit;

GPJ2CON &= ~(0xFF << 0);

GPJ2CON |= ((0x01 << 0) | (0x01 << 4));

GPJ2DAT |= (0xFF << 0);

GPH0CON &= ~(0xFF << 8);//接到外部中断2，3并于按钮相连

**while** (1)

{

key\_val = GPH0DAT & 0xC;

**if** (key\_val)

{

delay(500000);

key\_val = GPH0DAT & 0xC;

**if** (key\_val)

{

**if** (0x04 == key\_val)

GPJ2DAT ^= 1 << 1;

**else** **if** (0x08 == key\_val)

GPJ2DAT ^= 1 << 0;

}

}

}

**return** 0;

}

串口初始化P95

//初始化

GPA0CON &=~0xFF;

GPA0CON |=0x22;

ULCON0 = 0x0003;

UCON0 = 0x0045;

UFCON0 = 0;

UBRDIV0 = 34;//66000000/16/115200-1;

UDIVSLOT0 = 0xDDDD;

//发送数据

void putc(uchar c)

{

while(!(UTRSTAT0 & 1<<2 ));

UTXH0 = c;

return;

}

void puts(char \*str)

{

char \*p=str;

while(\*p)

{

putc(\*p++);

}

}

//接收

void char getc()

{

while(!(UTRSTAT0 & 1<<0))

return (URXH0);

}

int main(void)

{

char c;

init();

//清除，并设置输出功能引脚

GPJ2CON &= ~(0xFF << 0);

GPJ2CON |= ((0x01 << 0) | (0x01 << 4));

GPJ2DAT |= (0xFF << 0);

//点亮灯

GPJ2DAT = 0x0d;

puts(“点亮了从左开始第二个灯 1101”);

}

中断初始化：P108

初始化步骤：

(1) 配置GPIO引脚及其外部中断控制寄存器

(2) 选择中断类型（VICxINTSELECT）

(3) 清中断服务程序入口寄存器（VICxADDRESS）

(4) 设置相应中断的中断服务程序入口（VICxVECTADDR）

(5) 使能中断（VICxINTENABLE）

主程序：

(1) 启动程序

配置ARM工作模式、设计中断服务程序

(2) 初始化程序

初始化S5PV210中断控制器、配置GPIO引脚

(3) 主程序设计

按键控制LED

(4) 编写Makefile文件

编译生成目标文件

extern void IRQ\_handle(void);

void init\_key(void)

{  
 GPH0CON &= ~0xff;//gph0设置成中断;

GPH0=

}

定时器初始化：P128

定时器中断初始化：P128

AD参看ppt

系统时钟（PCLK）初始化。(上一届没考)

三、编程题：2道，一共45分

类似：

定时器+PWM+中断服务程序

串口+定时器+中断服务程序

注意在编程时，数字、字母的ASCALL码的使用方法，比如‘a’代表a的ASCALL码，也可以用97代表。

不要求会对PCLK时钟的配置。

但是所有的涉及到GPIO、串口、定时器、PWM、中断都要会初始化。

会提供编程涉及到的寄存器的英文版的数据手册截图（只保障编程题，不保障填空题）