**长沙理工大学考试试卷**

………………………………………………………………………………………………………

**课程名称**（含档次） 嵌入式系统（下） 课程代号

专 业 计算机科学与技术 层次（本、专） 本 考试方式（开、闭卷） 闭卷

# 填空题（10小题，每小题2分，共20分）

1.ARM核有两个指令集，分别是\_\_\_\_\_和Thumb指令集

答案：ARM指令集

2.嵌入式系统中SOC的中文释义是\_\_\_\_\_。

答案：片上系统

3.Ucos-II最多管理\_\_\_\_个任务。

答案：64

4.在Ucos-II中，在特定情况下，会出现低优先级任务比高优先级任务先运行的现象，被称为\_\_\_\_\_.

答案：优先级反转

5. Ucos-II是一个简洁、易用uC/OS-II是一个简洁、易用的 基于优先级的嵌入式\_\_\_\_\_\_多任务实时内核。

答案：抢占式

6.删除任务，是说任务将返回并处于\_\_\_\_\_\_\_，任务的代码不再被ucos-II调用。

答案：休眠状态

7．Ucos中，为保护任务之间的共享数据和提供任务之间的通信,提 供了操作系统任务间通信方法有:信号量、邮箱、\_\_\_\_,事件标志。

答案：消息队列

8.在Makefile中的命令必须要以\_\_\_\_键开始。

答案：Tab键

9.Linux环境下,rm命令可用于删除文件或目录，\_ \_\_命令可以移动文件和目录。

答案：mv命令

10. 为了提高的开发的效率，降低开发难度，提高重用性，经验告诉大家；\_\_\_是开发嵌入式系统软件的必由之路。

答案：基于嵌入式操作系统，采用模块化方式|采用模块化方式

填空题答案：

1.ARM指令集 2.片上系统 3.64 4.优先级反转 5. 抢占式

6. 休眠状态 7.消息队列 8.Tab键 9.mv(自己写的) 10.基于嵌入式操作系统，采用模块化方式|采用模块化方式

拓展：

1. 要建立一个概念：具有控制块的程序才是一个可以被系统所运行的任务。**程序代码、私有堆栈、任务控制块（TCB）**是任务的三要件。任务控制块提供了运行环境的存储位置。

# 选择题（15小题，每小题2分，共30分）

1、0X07&0X11的运算结果是

A、0x01 B、0x11 C、0x17 D、0x07

答案：A

2、以下ARM指令中，（）的源操作数采用了寄存器间接寻址方式。

A. MOVR0,#2

B. LDRR0,[R1]

C. BLSUB1

D. ADD R0,R1,R2,LSL#1

答案：B

3、以下哪个不是RISC架构的ARM微处理器的一般特点:( )

A、体积小、低功耗

B、大量使用寄存器

C、采用可变长度的指令格式,灵活高效

D、寻址方式灵活简单

答案：C

3拓、以下（）是RISC架构的ARM处理器的一般特点。

A、指令集中大部分为单周期指令

B、大量使用寄存器,指令执行速度更慢

C、RISC是复杂指令集计算机

D、寻址方式复杂,执行效率低

答案：A

4、嵌入式系统加电或复位后，所有的 CPU 通常都从某个由 CPU 制造商预先安排的地址上取指令。例如：对于S3C2410来说，Boot Loader会映射到\_\_\_\_\_\_地址处

A、0x0c000000

B、0x00000000

C、0xFFFFFF00

D、0x40000018

答案：B

5、在下列ARM处理器的各种模式中,\_\_\_\_模式有自己独立的R8-R14寄存器。

A、系统模式(System)

B、终止模式(Abort)

C、中断模式(IRQ)

D、快中断模式(FIQ)

答案：D

6、嵌入式系统由硬件部分和软件部分构成，以下（）不属于嵌入式系统软件。

A．系统内核

B．驱动程序

C．FPGA编程软件

D．嵌入式中间件

答案：C

C其实是芯片？？

7、在CPU 和物理内存之间进行地址转换时， ( ) 将地址从虚拟(逻辑)地址空间映射到物理地址空间。

A．TCB

B．MMU

C．CACHE

D．DMA

答案：B

8、在输入输出控制方法中，采用（）可以使得设备与主存间的数据块传送无需CPU干预。

A．程序控制输入/输出

B．中断

C．DMA

D．总线控制

答案：C

9、某系统以RS-232为接口,则该接口的最远传输距离为()

A、10m

B、15m

C、120m

D、1200m

答案：B

9拓、某系统以RS-232为接口,进行7位ASCII码字符传送,带有一位奇校验位和两位停止位,当波特率为9600时,字符传送率为\_\_A\_。

A.960 B.873 C.840 D.480

答案：A

解析：本题考查计算机系统方面的基础知识。

  某系统以RS-232为接口，进行7位ASCII码字符传送，带有一位奇校验位和两位停止位，那么传送一个字符共需二进制10位。所谓传输率就是指每秒传输多少位，传输率也称为波特率。按照题意，当波特率为9600时，字符传送率应为波特率除以传送一个字符的位数，故本题字符传送率为960。

10、Linux文件系统的目录结构是一棵倒挂的树，文件都按其作用分门别类地放在相关的目录中。现有一个外部设备文件，我们应该将其放在\_\_\_\_\_\_目录中。

A．/bin

B．/etc

C．/dev

D．/lib

答案：C

11、LINUX下，建立一个新文件可以使用的命令

A. chmod

B. more

C. cp

D. touch

答案：D

12、LINUX支持网络文件系统NFS,下列哪个命令实现了将位于192.168.1.4机器上的/opt/sirnfs目录挂载到本机/mnt/sirnfs下:()

A、mount -t nfs 192.168.1.4:/opt/sirnfs /mnt/sirnfs

B、mount -t nfs /mnt/sirnfs 192.168.1.4:/opt/sirnfs

C、mount nfs –t 192.168.1.4:/opt/sirnfs /mnt/sirnfs

D、mount nfs –t /mnt/sirnfs 192.168.1.4:/opt/sirnfs

答案：A

13、在使用GCC编译器的过程中，以下（）选项可用来指定生成的目标文件名

A . -c

B . -o

C . -S

D . -E

答案：B

14.在ARM Linux体系中,用来处理外设中断的异常模式是（）

A、软件中断(SWI)

B、未定义的指令异常

C、中断请求(IRQ)

D、快速中断请求(FIQ)

答案：C

15.嵌入式应用软件开发阶段不包括（）

A. 交叉编译和链接 B. 开发环境的建立 C. 联机调试 D. 应用程序模块

答案：D

答案：

1.A2.B3.C4.B 5.D 6.C

7.B**解析：**

A.TCB,线程控制块；PCB，进程控制块

B.MMU(Memory Management Unit) 内存管理单元，是\*\*\*处理器用来管理虚拟内存和物理内存寄存器的控制线路，同时也负责虚拟内存映射为物理内存等。

C.CACHE,感觉在这道题里应该指的是高速缓冲存储器。

D.DMA，（direct memory access）直接内存存储，传输数据从一个地址空间到另一个地址空间。

8.C



9.B 10.C 11.D 12.A

13.B

解析：**-c**

只激活预处理,编译,和汇编,也就是他只把程序做成obj文件

例子用法:

gcc -c hello.c

他将生成 .o 的 obj 文件

**-S**

只激活预处理和编译，就是指把文件编译成为汇编代码。

例子用法:

gcc -S hello.c

他将生成 .s 的汇编代码，你可以用文本编辑器察看。

**-E**

只激活预处理,这个不生成文件, 你需要把它重定向到一个输出文件里面。

例子用法:

gcc -E hello.c > pianoapan.txt

gcc -E hello.c | more

14.C 15.D

# 简答题（2小题，每小题5分，共10分）

1.阅读理解下列Ucos-II代码，请为各编号代码行添加注释。

void OSStart (void){ INT8U y; INT8U x; if (OSRunning == FALSE) { y = OSUnMapTbl[OSRdyGrp]; **1**

x = OSUnMapTbl[OSRdyTbl[y]]; **2** OSPrioHighRdy = (INT8U)((y << 3) + x);  **3**

OSPrioCur= OSPrioHighRdy; OSTCBHighRdy= OSTCBPrioTbl[OSPrioHighRdy]; **4**

OSTCBCur= OSTCBHighRdy; OSStartHighRdy(); **5**

}}

OSStart（）函数源码

void OSStart (void)

{

INT8U y;

INT8U x;

if (OSRunning == FALSE) {//参数检验，多任务尚未启动

**y = OSUnMapTbl[OSRdyGrp];**

//这几句通过uc/os设定最高优先级查找算法找出当前最高优先级

**最高优先级任务所在行**

**x = OSUnMapTbl[OSRdyTbl[y]];**

**最高优先级任务所在位**

**OSPrioHighRdy = (INT8U)((y << 3) + x);**

**还原为优先级**

OSPrioCur = OSPrioHighRdy;

//第一次开始多任务调度，最高优先级任务运行，当前优先级和最高优先级一样

**OSTCBHighRdy = OSTCBPrioTbl[OSPrioHighRdy];**

**//通过最高优先级超找优先级TCB表获得最高优先级TCB**

**通过上面两行代码将当前最高优先级的任务的优先级存放在OSPrioHighRdy变量中。然后通过此变量从存放任务控制块指针的数组OSTCBPrioTbl[]中获得该任务的任务控制块指针，并存放在指针变量OSTCBHighRdy中。**

OSTCBCur = OSTCBHighRdy;//第一次启动，二者一样

**OSStartHighRdy();//调用启动函数，实质上是将任务栈中保存的值弹回到CPU寄存器中，然后执行中断返回自指令，中断返回指令强制执行任务代码**

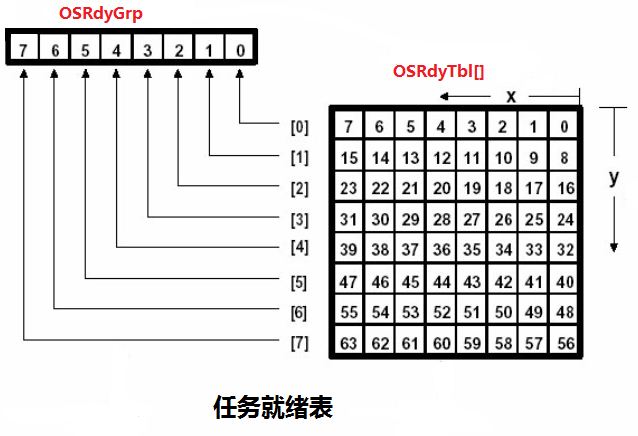
}

}

OSStartHighRdy()永远不返回到OSStart（）

当OSSTART执行完后OSRunning为真，多任务已经启动

他们3个用的都是INT8U的变量，也就是无符号的8bit变量。



2.简述Ucos-II任务代码的一般框架。

一个用C语言编写的任务(超循环结构的任务示意性代码):

void MyTask( void \*pdata )

{

for(;;)

{

可以被中断的用户代码;

OS\_ENTER\_CRITICAL(); //进入临界段(关中断)

不可以被中断的用户代码;

OS\_EXIT\_CRITICAL(); //退出临界段(开中断)

可以被中断的用户代码;

}

}

拓展：1.简述uC/OS-II的任务框架

答： void task\_xxx(void \*pData){

/\* 该任务的初始化工作 \*/

……

/\*进入该任务的死循环 \*/

while(1) {}

}

2、描述建立任务OSTaskCreate()的函数原型。

答：建立任务OSTaskCreate()的函数原型为：

INT8U OSTaskCreate (void (\*task)(void \*pd), void \*pdata, OS\_STK \*ptos, INT8U prio)

其中, task: 任务代码的指针;

pdata: 当任务开始执行时传递给任务的参数的指针;

ptos: 分配给任务的堆栈的栈顶指针; prio: 分配给任务的优先级。

3、试写出OSStartHighRdy( )函数的示意代码

void OSStartHighRdy() {

调用用户定义的OSTaskSwHook(); OSRuning = TRUE;

得到将要恢复运行任务的堆栈指针:

Stack pointer = OSTCBHighRdy->OSTCBStkPtr; 从新任务堆栈中恢复处理器的所有寄存器; 执行中断返回指令; ｝

4、试写出函数 OSCtxSw( )的示意性代码:

void OSCtxSw() {

保存处理器寄存器;

在当前任务的任务控制块中保存当前任务的堆栈指针:

OSTCBCur->OSTCBStkPtr = stack pointer;

OSTaskSwHook();

OSTCBCur=OSTCBHighRdy;

OSPrioCur=OSPrioHighRdy;

得到将要开始运行的任务的堆栈指针:

Stack pointer = OSTCBHighRdy->OSTCBStkPtr;

从新任务的任务堆栈中恢复处理器所有寄存器的值; 执行中断返回指令;

｝

5、试写出函数OSIntCtxSw()函数的示意性代码:

void OSIntCtxSw() {

调用用户定义的

OSTackSwHook();

OSTCBCur=OSTCBHighRdy;

OSPrioCur=OSPrioHighRdy;

得到将要重新执行的任务的堆栈指针:

Stack pointer = OSTCBHighRdy->OSTCBStkPtr; 从新任务堆栈中恢复所有寄存器;

执行中断返回指令; }

6、试写出函数 OSTickISR()函数的示意性代码:

void OSTickISR() {

保存处理器寄存器;

调用OSInitEnter( )或者OSInitNesting加1; if(OSInitNesting ==1) {

OSTCBCur->OSTCBStkPtr=stack pointer;

}

给产生中断的设备清中断; 重新允许中断; OSTimeTick();

OSInitExit();

恢复处理器寄存器;

执行中断返回指令; ｝

7、编写使任务进入就绪态代码:

OSRdyGrp |=OSMapTbl[prio>>3];

OSRdyTbl[prio>>3] |=OSMapTbl[prio&0x07];

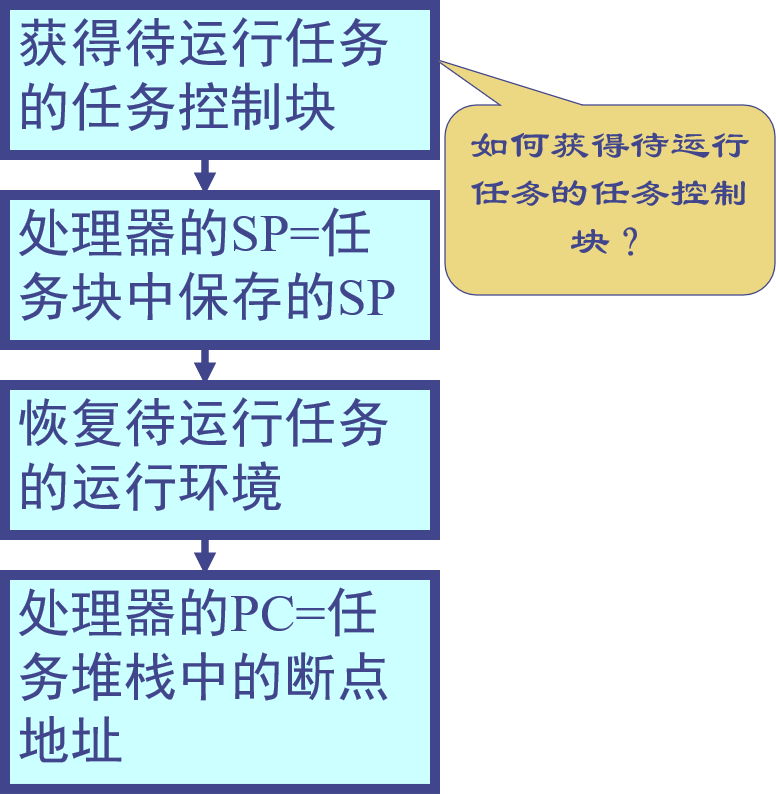
8、编写使任务脱离就绪态代码:

if((OSRdyTbl[prio>>3] &= ~OSMapTbl[prio &0x07])==0) OSRdyGrp &= ~OSMapTbl[prio>>3];

0、编写找到进入就绪态的优先级最高的任务，并计算其优先级值的代码:

y = OSUnMapTbl[OSRdyGrp]; x = OSUnMapTbl[OSRdyTbl[y]]; prio = y<<3+x;

9、任务切换的机理。



10. 什么是优先级反转？

答案：在可剥夺型内核中，当任务以独占方式使用共享资源时，会出现低优先级任务先于高优先级任务而被运行的现象，这种现象叫做任务优先级反转。

**根源：**使用信号量的任务是否能够运行是受任务的优先级别和是否占用信号量

两个条件约束的，而信号量的约束高于优先级别的约束。

**解决问题的办法之一：**使获得信号量任务的优先级别在使用共享资源期间暂时提升到所有任务最高优先级的高一个级别上，以使该任务不被其他的任务所打断，从而能尽快地使用完共享资源并释放信号量，然后在释放了信号量之后再恢复该任务原来的优先级别。

11.OSSemPend()

等待一个信号量， 即操作系统中的P操作，将信号量的值减1；

COSSemPend (OS\_EVENT \*pevent,INT16U timeout, INT8U \*err);

执行步骤:

1)如果信号量的计数值大于0， 将它减1并返回；

2)如果信号量的值等于0，则调用本函数的任务将被阻塞起来，等待另一个任务把它唤醒；

3)调用OSSched()函数，调度下一个最高优先级的任务运行。

12.OSSemPost()

发送一个信号量， 即操作系统中的V操作， 将信号量的值加1；

OSSemPost (OS\_EVENT \*pevent);

执行步骤:

1)检查是否有任务在等待该信号量，如果没有，将信号量的计数值加1并返回；

2)如果有，将优先级最高的任务从等待任务列表中删除，并使它进入就绪状态；

3)调用OSSched()，判断是否需要进行任务切换。

void OSTimeDly (INT16U ticks)

{#

if OS\_CRITICAL\_METHOD = = 3

OS\_CPU\_SR cpu\_sr;

#endif

if (ticks > 0)

{

OS\_ENTER\_CRITICAL( );

if ((OSRdyTbl[OSTCBCur->OSTCBY]&= ~OSTCBCur->OSTCBBitX) = = 0)

{

OSRdyGrp&= ~OSTCBCur->OSTCBBitY; //取消当前任务的就绪状态

}

OSTCBCur->OSTCBDly = ticks;//延时节拍数存入任务控制块

OS\_EXIT\_CRITICAL( );

OS\_Sched( ); //调用调度函数

}

}

# 编程题（2小题，每小题10分，共20分）

1.编写代码实现Ucos-II中的就绪表进行入表、出表、查表操作。

入表：

在程序中，可以用类似下面的代码把优先级别为prio的任务置为就绪状态：OSRdyTbl[prio>>3] | = OSMapTbl[prio&0x07];OSRdyGrp | =OSMapTbl[prio>>3];

出表：

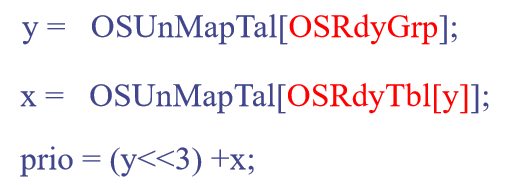
在程序中，可以用类似下面的代码把优先级别为prio的任务置为非就绪状态？OSRdyTbl[prio>>3] & = ~（OSMapTbl[prio&0x07]）;OSRdyGrp & =~（OSMapTbl[prio>>3] ）;

如果要使一个优先级别为prio的任务脱离就绪状态则可使用如下类似代码：

if((OSRdyTbl[prio>>3] &= ~OSMapTbl[prio&0x07]) == 0) OSRdyGrp&=~OSMapTbl[prio>>3];

查表：

在就绪表中计算机查找最高优先级别任务的代码：



2.Linux目录/home/arm下有一个C程序，它由几个单独的源文件组成，各源文件，又分别包含其他文件，如下图所示。

|  |  |
| --- | --- |
| 源码文件 | 包含文件 |
| Main.c | stdio.h，table.h |
| List.c | list.h |
| Sysmbol.c | symbol.h |
| Table.c | table.h |

网上参考答案：

hello:main.o list.o symbol.o table.o

arm-linux-gcc –o hello main.o list.o symbol.o table.o

main.o:main.c stdio.h table.h symbol.h list.h

arm-linux-gcc -c –o main.o main.c

list.o:list.c list.h

arm-linux-gcc –c –o list.o list.c

symbol.o:symbol.c symbol.h

arm-linux-gcc –c –o symbol.o symbol.c

table.o:table.c table.h symbol.h list.h

arm-linux-gcc –c –o table.o table.c

clear:

rm –f hello \*.o

0. arm-linux-gcc -o example example.c

不加-c、-S、-E参数，编译器将执行预处理、编译、汇编、连接操作直接生成可执行代码。-o参数用于指定输出的文件，输出文件名为example,如果不指定输出文件，则默认输出a.out

1. arm-linux-gcc -c -o example.o example.c

-c参数将对源程序example.c进行预处理、编译、汇编操作，生成example.o文件。去掉指定输出选项"-o example.o"自动输出为example.o,所以说在这里-o加不加都可以

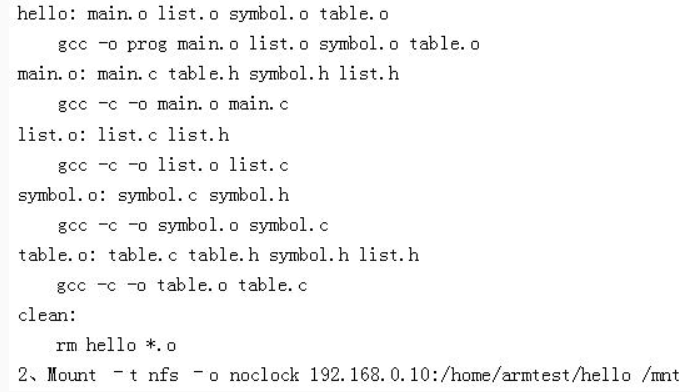
拓展：

（2） 在目标机上通过网络文件系统挂载/mnt到宿主机上的/home/arm目录。mount–t nfs–o nolock IP地址：/home/arm /mnt

已知C语言程序有主要程序模块prog.c、proc.h，其中调用了另一模块

subr.c、subr.h中的功能。试写出一个可将这两个模块编译成可执行文件pr1的Makefile。





2、用C语言编写一个源程序main.c，实现以下功能

a) 打开当前目录下的文件“text.txt”，如果没有则创建该文件，并使其具有读写属性 b) 编写一个makefile实现自动编译，生成可执行文件main

(1)

Void main(void) {

int fid;

fid = open(“./test.txt”,O\_RDWR|O\_CREAT);

if(fid==-1)

{

Printf(“open or create error \n”);

exit(0); }

Close(fid);

} (2)

objects = main.o exec = main all:$(objects)

gcc –o $(exec) $(objects) main.o:main.c

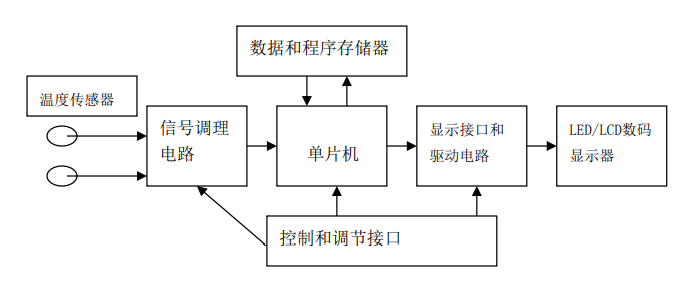
gcc –c main.c clean:

rm –r $(exec) $(objects)

# 设计题（1小题，每小题20分，共20分）

以下是一个具备设备温度检测及实时显示功能的应用系统硬件框图，请选择合适的嵌入式处理器和嵌入式操作系统进行开发设计并回答下列问题：

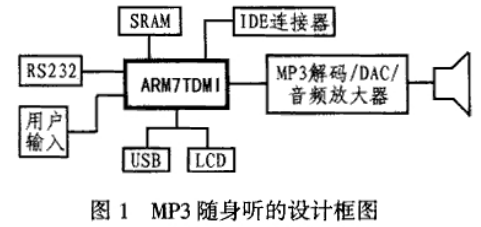
1. 分析说明该系统的开发与应用对社会的影响。（8分）
2. 简要说明系统开发所需环境工具及实现过程。（6分）
3. 请提供软件主流程图和关键代码。（6分）



在上述设计框图中，核心部分就是嵌入式处理器选择上要选择功能和处理速度满足需求的嵌入式处理器，最好内含A/D变换器，具有足够量的RAM和ROM程序存储体，设计和自制温度信号的输入、放大和切换调理部分电路,将数据转换后存储，通过相应的数码管或液晶显示屏幕完成数据显示、设计键 盘和调节借口，完成与系统的交互操作。其中根据硬件来编写和调试系统的软件部分是非常重要的工作量之一。

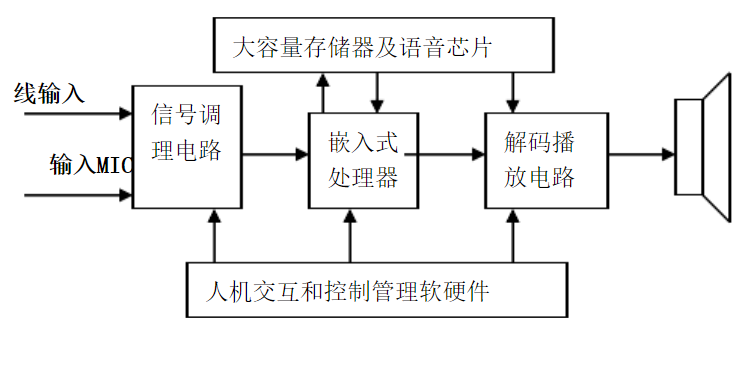
拓展：

1.现在流行的MP3音乐播放器是一个较为典型的嵌入式产品，其大致组成框图如下，请 选择简介其中五个模块的功能。



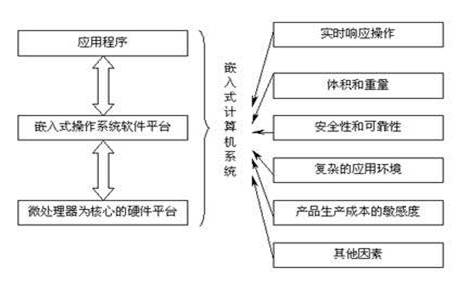
解答：（1）ARM7TDMI ARM7中最有特色，产销量大、性价比高而应用最广的一个芯片 （2）USB 接口，一般用于和PC机连接，完成音乐文件的传输。 （3） LCD 一般指一个液晶显示屏，显示和用户交互的信息。 （4） MP3解码+音频放大器，完成文件解码转换，音乐信号的处理。 （5） IDE为集成开发环境和调式接口。

2. 用嵌入式处理器开发一个嵌入式数码语音录放装置，请给出模块化设计框图,并对各部分作简单解释。



根据题意，画出模块框图如下。选择ARM7为嵌入式处理器，完成功能的控制和管理；解码器将存储的录音文件解码为语音信号；大容量存储器可选SRAM或FLASH存储器用于存放文件；信号调理电路将麦克风录入信号放大输入ARM7进行采集，还有简单按键和LCD显示器等完成人机交互。可用C或汇编编写系统控制管理和各功能程序。

**3.下图表示的是进行嵌入式系统产品开发的基础环境条件和进行开发时应当十分重视的技术要素。根据图上内容进行适当介绍。**



解答：嵌入式计算机系统产品开发的基础条件是要有一个软件嵌入式操作系统。或者需要一个使用的软件开发平台。并且具备所选CPU的硬件开发平台（如开发仿真机等。

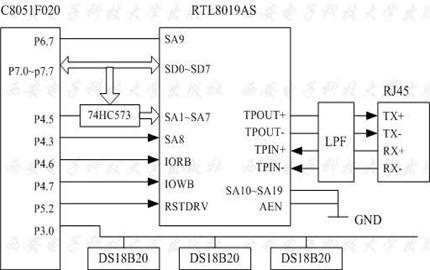
在设计与实现中始终讲究一些性能指标：操作处理的实时性，速度做够快。

必须注意压缩体积和重量和功耗等，因为嵌入式产品经常要做到便携。拿到现场工作。移动式产品安全并可靠工作非常重要。对于复杂的应用环境和其他一些因素，必须在设计初就应该计划考虑周到。要普及技术或还要考虑它的成本。

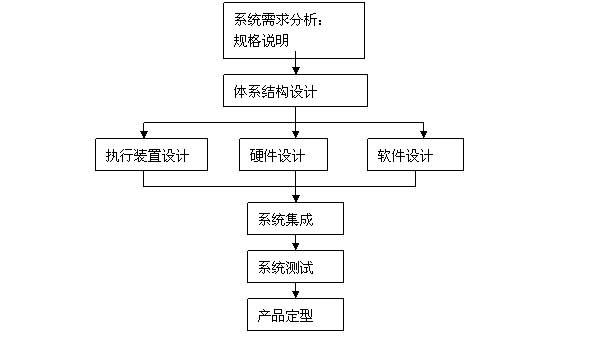
**4.根据所学知识，利用嵌入式单片机，设计一个多点温度采集系统，并可以接入网络进行温度数据的远程传输，画出硬件组成框图。并对其工作原理进行简述和说明。**

解答：根据题意，系统的硬件结构如图所示。处理器C8051F020的P7口采用复用方式与以太网控制器RTL8019AS的数据/地址线相接。P5.2与RTL8019AS的复位端相接，对其实现冷复位。RTL8019AS外接一个隔离LPF滤波器，通过RJ-45接口接入以太网。多个单总线温度传感器DSl8B20共享一条总线，由P3.0口进行控制。存储系统的使用方式是将C8051F020的XRAM作为输入/输出数据的内部缓冲区，将RTL8019AS内部的16 KB SRAM作为单片机的外部数据缓冲区，存储输入/输出以太帧队列。

（介质访问层主要由以太网控制器RTL8019AS来实现，其数据通信协议采用IEEE 802.3标准。软件可用C8051F020 C语言来编程实现）



**5、如图是一个一般嵌入系统设计的流程框图，请对流程作以恰当解释。**

****

解答：此流程图完整地描述了进行嵌入式系统和产品设计和开发的过程和步骤。我们按照该流图分阶段完成各步骤的工作。抓紧每个阶段的任务和重点，严把每个环节和保证工作质量，就能达到预想的目标。