CPU：寄存器组，指令，堆栈，中断，时钟、复位

ALC运算逻辑单元，寄存器组，控制单元（…）

运算逻辑单元的四个要素：操作数、运算、运算结果、标志位

PSR寄存器：存储单元产生的标志（条件码寄存器）（程序状态寄存器）ZNOC

存储与地址：

PC指针寄存器

栈：后入先出，push，pull顶端时最低位的地址

堆：一般是全局变量

堆栈溢出

轮询与中断

中断是由外部事件或者内部事件需要CPU立即处理，发生中断后，CPU停止正常流程，执行中断服务程序ISR，ISR结束后，CPU返回正常流程

中断允许、禁止：全局中断控制，CPU的CCR寄存器中一个特殊位，在复位后，全局禁止位是置起的，中断发生后，全局禁止位也被置起

中断标志位flag：每个中断源都有对应的中断标志，中断标志位将引发向CPU的中断请求，通过读写操作可以清除中断标志位

控制开关：CPSIE和CPSID

Register context：保存返回地址和寄存器上下文

中断服务子程（ISR）：参数和返回值都是void，如果允许中断的嵌套，那么要考虑清楚标志位

中断向量表：每一个中断源对应的中断函数地址

高优先级中断有优先相应

时钟是指令执行的基本事件间隔，时钟频率高，则CPU能力强

MCU内部是一个统一的时钟树，外设的时钟是从系统时钟分频得到的

锁相环级数，将外部较低频率的时钟提高成内部较高频率的时钟

20.97MHz

RAM保存非固定内容，掉电丢失

ROM保存固定内容，掉电不丢失

总线：数据线，地址线，控制线（通过总线，CPU访问片外储存器）

外设（peripheral）：相对于VPU的外部设备：

输入输出：DIO.ECT,PWM

模拟量处理：AD,DA

通信处理：SCI.SPI,I2C,HDLC,CAN

外设通过一系列寄存器来完成设置和工作（不同于CPU的寄存器）

外设：

IO

PTA0：PORT-A-0（端口PORT）

将IO口按照2的倍数合成一组

IO通常是双向的（输入或者输出）

IO引脚还与其他外设引脚复用

IO相关基础寄存器：

端口方向寄存器PDDR

端口数据输出寄存器PDOR

端口数据输入寄存器PDIR

端口其他寄存器PSOR,PCOR,PTOR

IO点灯：

PTD7-PTD0

1. 打开模块对应时钟
2. 将引脚设置为GPIO用
3. 设置IO端口为输入或者输出状态

变为高电位用或（|=0xFF）

变为低电位用与（&=~(0xFF)）

SIM\_SCGC5 = SIM\_SCGC5 |(1<<12);//打开时钟

每个引脚对应自己专门的功能配置寄存器 PORTx\_PCRn

PORTD\_PCR0=0x0100;//配置每个引脚的功能为IO

IO中断

嵌入式开发：

开发板-开发软件-模拟器-调试器-编程、烧写器

EVB开发板

0b100000

外设：

通讯：

串行、并行

同步、异步

点对点、总线

半双工、全双工

主从、对等

单端、差分

并行比串行快，现在更高通讯采用串行

点对点与总线，USB是总线通讯

接收端与发射端是否始终同步，是则是同步，反之为异步

半双工为收和发的双方在单位时间内只有一方可以完成通讯

全双工双发可以同时收发信息

USB通讯是主从通讯

单端信号与差分信号，现在多用差分信号

UART：通用异步串行通讯（点对点、全双工）

波特率：每秒钟传递多少个比特

LSB：最低位

MSB：最高位

9600,8N1

过采样

奇偶校验：Even Partity，Odd Partity

电平转化

UART寄存器编程：

1. 打开UART模块和引脚组时钟
2. 指定引脚功能为UART
3. 设置UART参数

UART0,UART1,UART2

UART有自己的寄存器

BDL和BDH

IO的中断：

打开终端：

1. 打开中断开关
2. 配置中断源，确定什么时候发生中断
3. 写好中断服务子程
4. 将中断服务子程填入中断向量表

asm(“CPSIE i”);//确认开启全局中断

NVIC\_ISER |=0x40000000;//开启PORTA中断

PORTA\_PCR1=0x0A0102;//假定按键对应PTA1，下降沿触发中断

清除中断标志位

关联中断与中断服务子程：需要找到中断向量表中对应的函数名称

定时器编程：

1. 打开中断

PWM：

脉冲调制模块

当前计数：TPMx\_CNT

周期设定：TPMx\_MOD

占空比设定：TPMx\_CnV

独立的三个PWM模块，每个模块有6个通道，每个模块的所有通道周期一定，每个通道可以单独设置工作模式和占空比

SPI通讯：4线同步串行通讯接口（一对一或者一对多）（同步，全双工，主从）

MOSI：主出从入，主设备向从设备发出信号

MISO：主入从出，从设备向主设备发出信号

SCK：由主设备控制产生的SPI工作时钟，每个SCK周期完成1个bit的传输

SS：从设备选择端，当从设备收到该端为低电平，设备有效，对于主设备，该端平时为高电平，可在有效传输周期内给出低电平选通