# 1.heap(堆)和stack（栈）的区别

heap是堆，stack是栈

stack的空间由操作系统自动分配/释放，heap上的空间手动分配/释放。

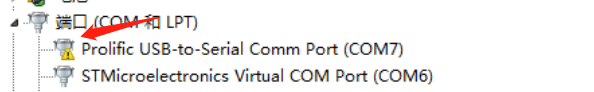
stack空间有限，heap是很大的自由存储区

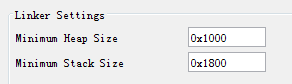
C中的malloc函数分配的内存空间即在heap上，C++中对应的是new操作符。

程序在编译期对变量和函数分配内存都是在stack（栈）上进行，且程序运行过程中的函数调用时参数的传递也在栈上进行。

# 2.STM32CebeMX 配置USB接口问题：

如果编译下载代码到开发板后，串口显示黄的感叹号，应调整堆栈空间





# 3.float类型导致变量无法获取数据（unavailable）问题：

解决方案：添加头文件#include"arm\_math.h"

（IAR工程）打开Options --->C/C++Compiler--->Preprocessor选项下找到Defined symbols添加下面四行代码

ARM\_MATH\_CM4

\_\_FPU\_PRESENT=1

\_\_FPU\_USED=1

\_\_ICCARM\_\_

# 4.一个iic总线上能挂载多少个从机？

有IIC地址决定，8位地址，减去1位广播地址，是7位地址，2^7=128，但是地址0x00不用，那就是127个地址， 所以理论

上可以挂127个从器件。

# 5.BSRR，BSR,ODR,IDR寄存器

BSRR 只写寄存器：[color=Red]既能控制管脚为高电平，也能控制管脚为低电平。

对寄存器高 16bit 写1 对应管脚为低电平，对寄存器低16bit写1对应管脚为高电平。写 0 ,无动作

00000000 00000000 00000000 00000000

1(左移13位)

1(然后左移16位进行写1操作)

对寄存器高 16bit 写1 对应管脚为低电平

#define MTEN\_LOW (GPIOC->BSRR = ((0x00000001U<<13)<<16))

#define MTEN\_HIGH (GPIOC->BSRR = (0x00000001<<13))

GPIOB->BSRR = 0x01就是把GPIOB port 0升为高电平

GPIOB->BRR = 0x01就是把GPIOB port 0降为低电平

#define MTDIR\_LOW (GPIOC->BSRR = ((0x00000001U<<14)<<16))

#define MTDIR\_HIGH (GPIOC->BSRR = (0x00000001<<14))

IDR读取引脚状态的寄存器，是32位的，但是只有低16位有效

例：判断PORTC的11位是否为低

if((GPIOC->IDR&1<<11)==0)

｛

//用户代码;

｝

#define MTCLK\_GET (GPIOC->IDR & (0x00000001<<15))

#define MTCLK\_LOW (GPIOC->BSRR = ((0x00000001U<<15)<<16))

#define MTCLK\_HIGH (GPIOC->BSRR = (0x00000001<<15))

GPIOA->ODR|=1<<8;//设置PA8输出1,led灭

GPIOA->ODR&=0<<8;//设置PA8输出0,led亮

#define MTCLK\_TOGGLE (GPIOC->ODR = (0x00000001<<15))

IDR是查看引脚状态寄存器

这个寄存器是32位的，所以定义的时候还是得定义为32位的，但是他只有低16位有作用。

格式

#define WARM\_KEY\_PIN (GPIOC->IDR&(1<<5)) ---->C5

1<<5表示：1左移5位（0000 0001---->0010 0000）

(endsPort[i]->IDR & endsPin[i])

# 6.tim定时器时间周期计算

（（1+分频系数）/（时钟晶振频率））\*（1+Period）=时间T

(1+83)/168\*(1+9)=5\*0.000001(s)=0.005ms

f=1/t

# 7.制作gerber文件

文件-->制造输出-->gerber file--> --- -->OK---文件-->导出-->gerber

文件-->制造输出-->NC Drill --- -->OK---文件-->导出-->gerber

详细步骤请参照C:\Users\Administrator\Desktop\我的文档\picture\gerber\_file\_get

# 8.PCB各层含义

1.阻焊就是盖油

---------(有空需要补充)

8.1.AD18区域性挖空板子步骤：首先使用画笔画出要挖空的区域形状，然后选中这个形状边框，最后点击选项工具-->装换--->从选择的元素创建非铺铜区域

9.ad18重新定义板子类型问题

需要先设置视图模式 视图---->模式1---->设计--->板子形状--->定义板子类型

修改移动栅格具体G

测距Ctrl + m

快速查找未连接得网络

显示未连线步骤

在左下角“PCB”框中上方下拉菜单“Nets”

接下来选择“<All Nets>”

在下面看到“Un-Routed”不为0的就是未连接

在前面打勾即可看到

然后进行二次检测，工具--->设计规则检查

# 9-1.怎样画不同形状的PCB板（例如圆形）

首先在keep-out-layer层画出一个圆形，然后点击设计选项-->板子形状--->按照选择对象定义（记得要先选中你画的图形哦）

# 9-2.keep-out-layer不能画外形问题？

新版本AD软件对禁止布线层进行了改进，操作步骤。选中KEEP-LAYER层，点击放置（P）-->Keepout（K）

10.浏览器快捷方式

CTRL+D添加收藏

CTRL+H历史记录

# 11.怎样快速清除IAR中的断点?

VIEW有个breakpoint 点一下就知道了

# 12.欧姆定律

u=ir;

v1=ir1;

v2=ir2;

v1/v2=r1/r2;

# 13.数组指针和指针数组

13.1.指针数组：首先它是一个数组，数组的元素都是指针，数组占多少个字节由数组本身的大小决定，每一个元素都是一个指针，在32 位系统下任何类型的指针永远是占4 个字节。它是“储存指针的数组”的简称。

13.2.数组指针：首先它是一个指针，它指向一个数组。在32 位系统下任何类型的指针永远是占4 个字节，至于它指向的数组占多少字节，不知道，具体要看数组大小。它是“指向数组的指针”的简称。

14.TIM定时器算法

((1+TIM\_Prescaler )/72M)\*(1+TIM\_Period )=((1+2)/72M)\*(1+65535)=0.00273秒

=366.2Hz

72M/(1+2)/(1+65535)=366.2hz=0.00273s

15.左移和右移

5<<1 即：0101左移一位1010 扩大原来的二倍

5>>1 即: 0101右移一位0010 缩小原来的二倍（取整）

# 16.异步和同步，阻塞和非阻塞

你打电话问书店老板有没有《分布式系统》这本书，如果是同步通信机制，书店老板会说，你稍等，”我查一下"，然后开始查啊查，等查好了（可能是5秒，也可能是一天）告诉你结果（返回结果）。

而异步通信机制，书店老板直接告诉你我查一下啊，查好了打电话给你，然后直接挂电话了（不返回结果）。然后查好了，他会主动打电话给你。在这里老板通过“回电”这种方式来回调。

所谓阻塞方式的意思是指, 当试图对该文件描述符进行读写时, 如果当时没有东西可读,或者暂时不可写, 程序就进入等待 状态, 直到有东西可读或者可写为止。而对于非阻塞状态, 如果没有东西可读, 或者不可写, 读写函数马上返回, 而不会等待 。

# 17.memcpy函数使用方法

uint\_8 sbuf[10]="we";

uint\_8 buf[10];

memcpy(buf,sbuf,sizeof(sbuf));

解释为:将sbuf的内容复制到buf中

补充：高电压总是往低电压流

# 18:枚举变量enum使用方法

enum gNum{

one = 0,

tow,three

};

gNum zero = one;

enum gBuf {six};

gNum 就成为了一个枚举类型，形同与int

# 19.windows cmd命令

打开cmd:windows+r

删除一个目录rd /s dir

删除一个文件del /p .txt

打开远程桌面mstsc

关机命令shutdown -s -t xx （xx是关机延时时间 /秒；xx可以是00，即时关机。有时候windows提示n分钟后关机，但实际是你设定的时间）

输入taskmgr，按回车键即可快速打开任务管理器

输入control命令快速打开控制面板

输入ifconfig查看本地IP地址

# 20. ESP8266使用笔记

SerialNet(透传)模式将本地异步串口通信转换成基于TCP/UDP协议的网络通信。

Esp8266

# 21.怎样把元件放在pcb的背面

选中元件，查看属性，修改所在层，就可以放在相应层上了

1.二极管反向截至，只允许电流单向流过

2.boot0和boot1可以做一种启动方式，在程序烧写时要进行设置短接高低电平

10uf钽电容有标记的为正极，无标记的接负极

# 22.typedef用法

例如 typedef int size; size a=0;这个size就是int的别名

typedef struct node{}node\_n;

typedef struct node A;

# 23.步进电机算法

例如8细分，步进角1.8°，一圈需要1600个脉冲，转360°,一个脉冲转0.225°（1.8/8=0.225°）

减速比27:1和转子减速比3:1，所以转一圈所需脉冲1600\*27\*3\*0.225=29160

29160个脉冲360°

29160/90 = 324

表示一度用324的脉冲

知识点1：直接对数据块（内存地址）进行操作

\*((int\*)1245024) = adc;

\*(uint16\_t\*)&buf[0] = adc;

知识点2：代码运行空间是栈空间，堆空间用来分配给变量的空间，即：不是用特定函数申请的空间（栈）在程序运行完后会被析构掉。特定函数例如malloc

知识点2：RCC时钟源

在STM32系统中，共有五个时钟源，分别为HSE、HSI、LSE、LSI、PLL。HSI和LSI为片内RC振荡器，

HSI为8MHZ而LSI为32Khz(30-60Khz之间);HSE和LSE为外部时钟源;PLL则需要HSE和HSI来提供时钟。

HSE:外部高速时钟

可通过OSC\_IN输入提供外部时钟源

HSI：内部高速时钟

该时钟通过内部8MhzRC振荡器产生，可以直接用做系统时钟

LSE:外部低速时钟

振荡器是一个32.768Khz的晶体/陶瓷振荡器，可以像RTC提供精准时钟源

LSI:内部低速时钟

知识点3：分频器的作用

通常用于单片机定时器，主要减小微控制器的时钟频率。

用网络助手发十六进制数时，要先以十进制形式在发送区输入十六个0，然后在转换成十六进制，每一位分别是buf[0]buf[1]----

1.位操作

sbuf[28] |= 0x01;0000 0001 //保留最低位，并存入sbuf

sbuf[28] |= 0x02;0000 0010

sbuf[28] |= 0x04;0000 0100

sbuf[28] |= 0x08;0000 1000

sbuf[28] |= 0x10;0001 0000

全双工（Full Duplex）是指在发送数据的同时也能够接收数据，两者同步进行，这好像我们平时打电话一样，说话的同时也能够听到对方的声音。

目前的网卡一般都支持全双工。

半双工（Half Duplex），所谓半双工是指一段时间内，只有一种动作发生，例如一条窄路，而只有一辆车可以通行，当当前有两辆车相对时，在这

种情况下，只有一辆车先开，等到另一辆车的头部再开，这个例子生动地说明了原始半双工。原因。早期对讲机、早期集线器等设备都是基于半双工

产品的。随着技术的不断进步，半双工会逐渐退出历史舞台。