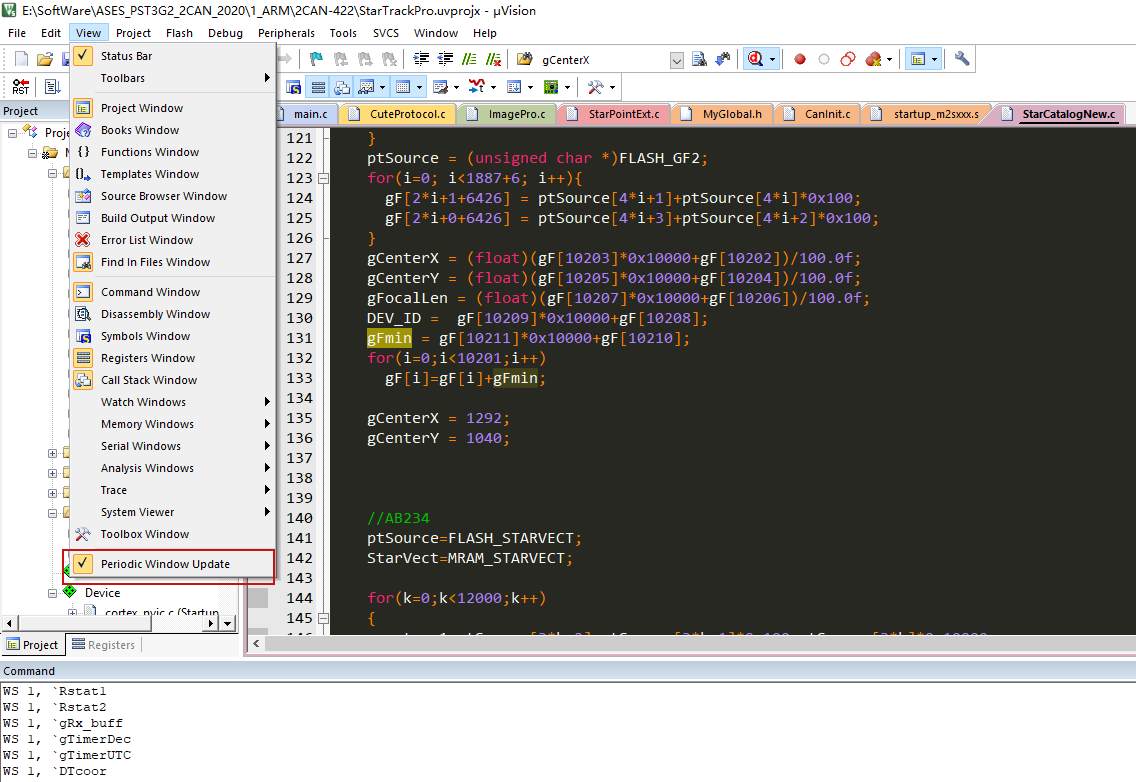
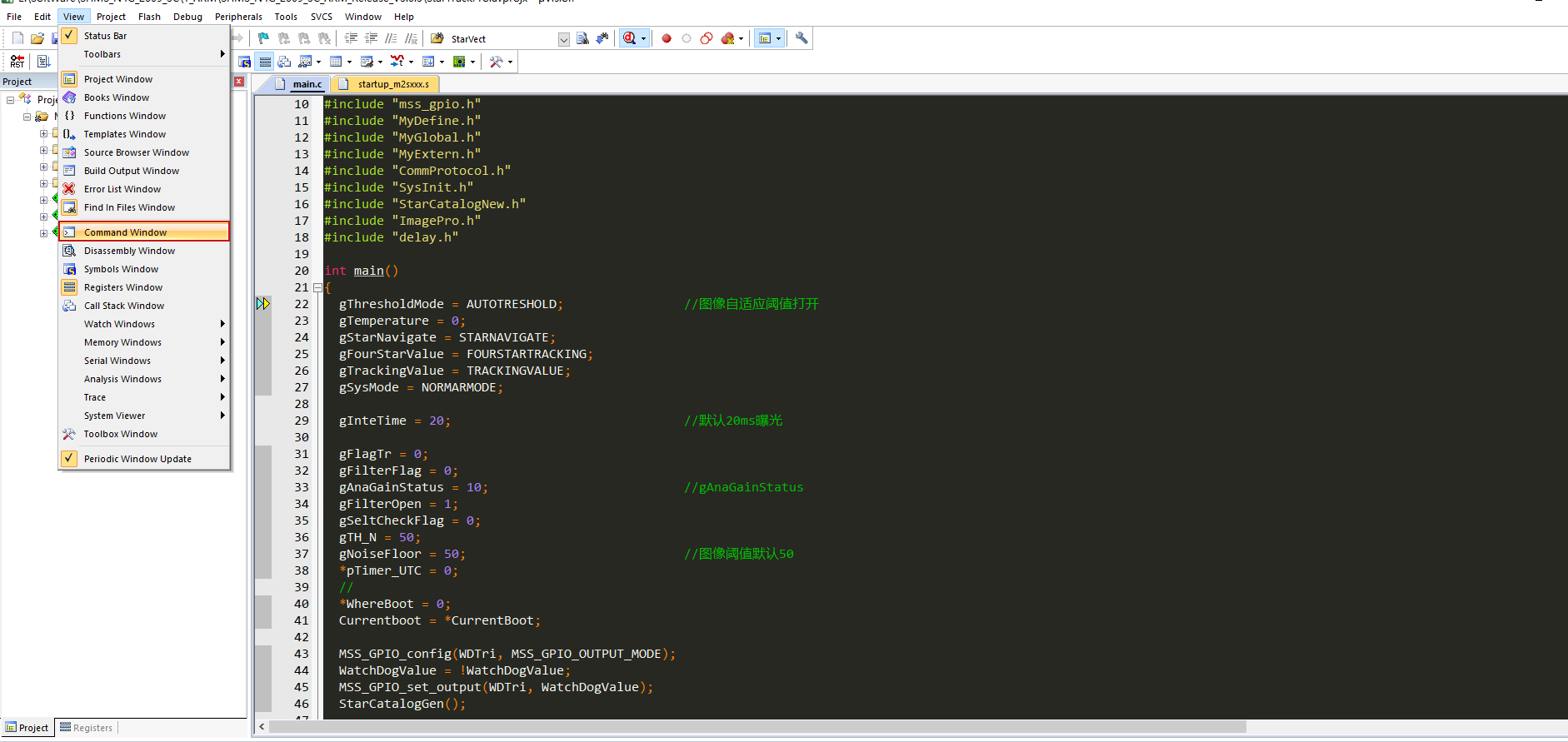
1. 嵌入式软件编程
   1. keil debug时watch窗口和memary窗口数据不能实时更新

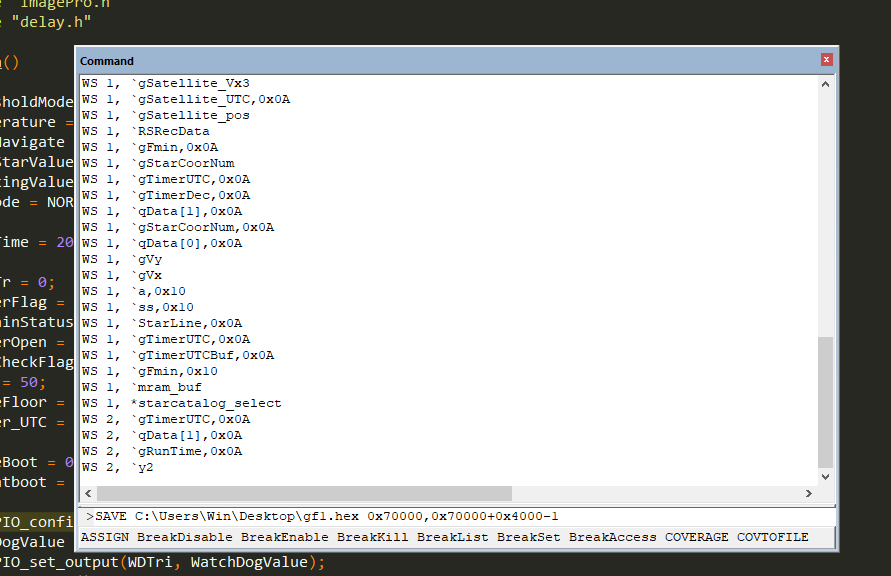
debug模式下勾选View->Periodic Window Update



* 1. Keil debug模式下读取内存数据保存为hex文件或txt文件

debug模式下勾选View->Command Window ,在弹出的命令窗口输入命令





导出数据使用SAVE 导入数据使用Load

例如：

SAVE C:\Users\Win\Desktop\cal.txt 0x70000,0x70000+1000 //读内存

LOAD

* 1. M2S090芯片串口使用Printf函数的配置方法

printf的使用 需要更改配置文件retarget.c

#ifdef MICROSEMI\_STDIO\_THRU\_MMUART0

static mss\_uart\_instance\_t \* const gp\_my\_uart = &g\_mss\_uart0;

#else

static mss\_uart\_instance\_t \* const gp\_my\_uart = &g\_mss\_uart1;

#endif

if(!g\_stdio\_uart\_init\_done)

{

MSS\_UART\_init(gp\_my\_uart, MICROSEMI\_STDIO\_BAUD\_RATE,

MSS\_UART\_DATA\_8\_BITS | MSS\_UART\_ODD\_PARITY);

g\_stdio\_uart\_init\_done = 1;

}

* 1. 仿真器

ULINK是KEIL公司开发的仿真器，专用于KEIL 平台下使用，ADS, IAR 下不能使用

JLINK 是通用的开发工具，可以用于KEIL ，IAR ，ADS 等平台 速度，效率，功能均比ULINK强

* 1. 负数用补码输出

if(temp\_buf>0)

gTemperature = temp\_buf;

else

gTemperature = ~(-temp\_buf)+1;

* 1. KEIL编译警告

KEIL MDK ARM中

unsigned int value2 = 0x80000000;

unsigned int value4 = 2147483648;

value2编译时不会产生警告，而value4就产生警告 warning:#1134-D: literal terated as "long long"

解决方法：

关键时2147483648 是一个字面常量，类型是int,而不是unsigned int.

上面这个语句，需要先将int类型，转换成unsigned int类型，再赋值给value4;

在转换过程中，因为这个数值超过32位int类型的表示范围，所以会出现警告。

unsigned int value2 = 0x80000000;

0x80000000;字面类型是unsigned的，所以在32位的表示范围内

解决方法： unsigned int value4 = 2147483648u;

* 1. 枚举型定义错误状态

typedef enum nvm\_status

{

NVM\_SUCCESS = 0,

NVM\_PROTECTION\_ERROR,

NVM\_VERIFY\_FAILURE,

NVM\_PAGE\_LOCK\_ERROR,

NVM\_WRITE\_THRESHOLD\_ERROR,

NVM\_IN\_USE\_BY\_OTHER\_MASTER,

NVM\_INVALID\_PARAMETER

} nvm\_status\_t;

1. MATLAB软件编程
   1. MATLAB中的取整函数

floor(x) 向下取整 例：floor(1.2)=1 floor(2.5)=2 floor(-2.5)=-3;

ceil(x) 向上取整 例：ceil(1.2)=2 ceil(2.5)=3 ceil(-2.5)=-2;

round(x) 取最接近的整数，如果小数部分是0.5，则向绝对值大的方向取整

例：round(1.2)=1 round(2.5)=3 round(-2.5)=-3

fix(x) 向0取整 例：fix(1.2)=1 fix(2.5)=2 fix(-2.5)=-2

* 1. 数组 cell

对于单元数组C，C(m,n)是指第m行第N列的单元，而C{m,n}是指单元数组中第m行第n列单元的内容

* 1. 字符串类型与数字类型转换函数

abs 字符串转换成ASCII

dec2hex 十进制数转换成十六进制字符串

fprintf 把格式化的文本写到文件中或显示屏上

hex2dec 十六进制字符串转换成十进制数

hex2num 十六进制字符串转换成IEEE浮点数

int2str 整数转换成字符串

lower 字符串转换成小写

upper 字符串转换成大写

num2str 数字转换成字符串

str2num 字符串转换成数字

sprintf 用格式控制转换成字符串

* 1. MATLAB中的关系操作符

< 小于

<= 小于等于

> 大于

>= 大于等于

== 等于

~= 不等于

* 1. MATLAB中的数值类型分类

整型 :

int8 有符号8位整数 -2^7~2^7-1

uint8 无符号8位整数 0~2^8-1

int16 有符号16位整数 -2^15~2^15-1

uint16 无符号16位整数 0~2^16-1

int32 有符号32位整数 -2^31-2^31-1

uint32 无符号32位整数 0~2^32-1

int64 有符号64位整数 -2^63~2^63-1

uint64 无符号64位整数 0~2^64-1

注：除int64和uint64类型外的所有整数类型，都可以进行数学运算

单精度浮点型：single

双精度浮点型：double MTALAB中默认的数值类型

注：在未加说明与特殊定义时，MATLAB对所有数值按照双精度浮点数类型进行存储和操作

* 1. 矩阵形状信息的查询函数

ndims n=ndims(A) 获取矩阵的维数

size [m,n]=size(A) 获取矩阵在各个维上的长度

length n=length(A) 获取矩阵最长维的长度

numel n=numel(A) 获取矩阵元素的个数

* 1. 初始化矩阵

A = zeros(m,n); %初始化一个m×n数值为0的矩阵

A = ones(m,n); %初始化一个m×n数值为1的矩阵

A = ones(m,n)\*K; %初始化一个m×n数值为K的矩阵

* 1. 图片像素位置查看

impixelinfo

* 1. norm()函数
  2. GUI -- 进度条的使用

doc waitbar 查看进度条的使用

例程

h = waitbar(0,'Please wait...');

steps = 1000;

for step = 1:steps

% computations take place here

h = waitbar(step / steps)

end

waitbar(1,h,'program done!')

* 1. GUI 串口部分

function COM\_OPEN\_Callback(hObject, eventdata, handles)

% hObject handle to COM\_OPEN (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

%数据初始声明

global invalid\_num; %无效帧数

global valid\_num; %有效帧数

global rec\_num; %接收帧数

global filename;

global LISTBOXNUM;

global a4Q;

global NUM\_SEND;

global Flag\_CH;

Flag\_CH=0;

%数据初始化

NUM\_SEND=0;

invalid\_num=0;

valid\_num=0;

rec\_num=0;

LISTBOXNUM=0;

a4Q=[];

global s;

global t;

global va;

COM=get(handles.COM,'value');

switch COM

case 1

s = serial('COM1');

case 2

s = serial('COM2');

case 3

s = serial('COM3');

case 4

s = serial('COM4');

case 5

s = serial('COM5');

case 6

s = serial('COM6');

case 7

s = serial('COM7');

case 8

s = serial('COM8');

case 9

s = serial('COM9');

case 10

s = serial('COM10');

case 11

s = serial('COM11');

case 12

s = serial('COM12');

case 13

s = serial('COM13');

case 14

s = serial('COM14');

case 15

s = serial('COM15');

case 16

s = serial('COM16');

case 17

s = serial('COM17');

case 18

s = serial('COM18');

case 19

s = serial('COM19');

case 20

s = serial('COM20');

case 21

s = serial('COM21');

case 22

s = serial('COM22');

case 23

s = serial('COM23');

case 24

s = serial('COM24');

case 25

s = serial('COM25');

case 26

s = serial('COM26');

case 27

s = serial('COM27');

case 28

s = serial('COM28');

case 29

s = serial('COM29');

case 30

s = serial('COM30');

end

Baud=get(handles.Baud,'value');

switch Baud

case 1

s.baudrate=115200;

case 2

s.baudrate=9600;

case 3

s.baudrate=19200;

case 4

s.baudrate=38400;

end

JIAOYAN = get(handles.JIAOYAN,'Value');

switch JIAOYAN

case 1

s.parity = 'odd';

case 2

s.parity = 'even';

case 3

s.parity = 'none';

end

s.stopbits =1;

s.databits =8;

s.timeout=0.5;

s.InputBufferSize = 1024;

s.OutputBufferSize =1024;

s.ReadAsyncMode = 'continuous';

if va==1

s.BytesAvailableFcnMode='byte';

s.BytesAvailableFcnCount=53;

s.BytesAvailableFcn={@ser\_callback,handles};

handles.scom=s;

end

%%打开串口

try

fopen(s);

catch

msgbox('所选串口无法打开！','replace');

end

set(handles.COM\_OPEN,'enable','off');

function COM\_CLOSE\_Callback(hObject, eventdata, handles)

% hObject handle to COM\_CLOSE (see GCBO)

% eventdata reserved - to be defined in a future version of MATLAB

% handles structure with handles and user data (see GUIDATA)

global s;

try

fclose(s);

delete(s);

catch

msgbox('所选串口无法关闭！','replace');

end

set(handles.COM\_OPEN,'enable','on');

* 1. 自动创建文件夹

例程：

for i=1:10

filename = ['mkdir A-',num2str(i)];

system(filename); %在当前文件夹下建立文件夹

end

* 1. 上位机批量显示

MATLAB代码参考

for i = 1:46

idx = num2str(i);

tagx = ['tag' idx]; %空格

set(handles.(tagx),'String',num2str(recdta(i),'%02X '));

end

* 1. mem文件处理

对于mem文件读取，如星表文件，每个字节没有完全8位对齐，使用cell2mat() 整体转会报错，这时候可以使用for循环单独进行转换即可。

* 1. Hex转浮点型

C=B(1)\*2^24+B(2)\*2^16+B(3)\*2^8+B(4);

D=dec2hex(C);

E=typecast(uint32(hex2dec(D)),'single');

* 1. 图像显示

PictureName='1.jpg';

Picture=imread(PictureName);

Picture=Picture(:,:,1);

imshow(Picture);

hold on;

fLenth=29.111;

gpel=0.0069;

gCMOSXLEN=1498;

gCMOSYLEN=1498;

r=tand(20/2)\*fLenth/gpel;

xc = gCMOSXLEN/2;

yc = gCMOSYLEN/2;

theta = linspace(0,2\*pi);

x = r\*cos(theta) + xc;

y = r\*sin(theta) + yc;

plot(x,y,'g');

hold on;

starEx=cell2mat(StarPosFram(2,3));

starCap=cell2mat(StarPosFram0(1,14));

[m,n]=size(starCap);

for j=1:m

plot(starCap(j,1),starCap(j,2),'g--o','MarkerSize',8);

text(starCap(j,1)+30,starCap(j,2)+30,num2str(starCap(j,7)),'color','g');

text(starCap(j,1),starCap(j,2)+20,num2str(j),'color','g');

end

[k,j]=size(starEx);

for i=1:k

plot(starEx(i,1),starEx(i,2),'b--\*','MarkerSize',5);

text(starEx(i,1),starEx(i,2)+40,num2str(i),'color','b');

end

* 1. plot图像设置

1. 消除警告 warning('off');
2. 设置图像背景值为白色，方便写文档 set(0,'defaultfigurecolor','w');
3. 设置图例 legend('xxxx', 'xxxx');
4. 设置标题 title('xxxxxxxxxxxxxxxx');

（5）随时间变化 plot(UTC,DataMat(:,20:23));

datetick('x','yyyy-mm-dd HH:MM:SS','keepticks');

* 1. 保存图片并且保存到指定的文件夹

saveas(gcf,'03.bmp','bmp');

path1=[str1,'\',str2,];

movefile('03.bmp',path1);

* 1. mex -setup报错



解决方法

1、下载配置MinGw C/C++ ;配置环境变量，CMD可查gcc –v可查

2、setenv('MW\_MINGW64\_LOC','C:\mingw64')（配置自己的路径）

生成动态链接库

mcc -W cpplib:StarTracker -T link:lib DIF\_422.m –C

* 1. 提取字符串中数字正则表达式

regex='\d+';

matches=regexp(str,regex,'match');

* 1. 字符串倒序

str=reverse(str);

* 1. 按位取反

bitcmp(uint8(a))；

注意要加uint8、uint16

1. 计算概念
   1. 角度换算

1°（度）= 60′（角分）= 3600″ （角秒）

1弧度 = 180/Π 度 = 57.3°

1度 = Π/180 弧度

Π = 180°

* 1. 地球自转

地球自转速度为7.292\*10^-5 弧度/s = 7.292\*10^-5\*180/Π\*3600 = 15.0408 ″/s

* 1. 星敏领域的概念

1）赤经、赤纬

2）欧拉角

3）J2000坐标系

4）惯性坐标系

5）视场角计算

6）像素阵列

* 1. 星敏算法

软件方面根据测量精度和处理速度的要求选择合适的亚像元质心确定算法。选择基于四元数的姿态确定算法及最优估计算法进行三轴姿态估计和滤波，并根据系统参数对算法进行优化。采用了基于导航星域和k矢量相结合的主星星图识别方法以及全天自主模式下的递推的星图识别算法和跟踪模式下的递推星图识别算法。

* 1. channel link

（高速差分信号线 DOUT+ ,DOUT-）

* 1. 遮光罩

遮光罩的设计由公司自主设计完成。遮光罩采用吸收型遮光罩，内表面采用高吸光比的黑色阳极化涂层。

* 1. 宇宙速度

第一宇宙速度，指物体在地面附近绕地球做匀速圆周运动的速度叫做第一宇宙速度。在一些问题中说，当某航天器以第一宇宙速度运行，则说明该航天器是沿着地球表面运行的。按照力学理论可以计算出v1=7.9 km/s

第二宇宙速度，当航天器超过第一宇宙速度v1达到一定值时，它就会脱离地球的引力场而成为围绕太阳运行的人造行星，这个速度就叫做第二宇宙速度，亦称脱离速度。所谓摆脱地球束缚，就是几乎不受地球引力影响，这与处于离地球无穷远点的位置得情况等价。这里要注意，由于月球还未超出地球引力的范围，故从地面发射探月航天器，不需要达到第二宇宙速度v2，实际上其初始速度不小于10.848 km/s 即可。

第三宇宙速度，从地球表面发射航天器，飞出太阳系，到浩瀚的银河系中漫游所需要的最小发射速度，就叫做第三宇宙速度。按照力学理论可以计算出第三宇宙速度V3=16.7公里/秒。需要注意的是，这是选择航天器入轨速度与地球公转速度方向一致时计算出的V3值；如果方向不一致，所需速度就要大于16.7公里/秒了。可以说，航天器的速度是挣脱地球乃至太阳引力的唯一要素，只有火箭才能突破该宇宙速度。

三大宇宙速度与人造卫星的发射

1）当v<v1时，被发射物体最终仍将落回地面；

2）当v1≤v<v2时，被发射物体将环绕地球运动，成为地球卫星；

3）当v2≤v<v3时，被发射物体将脱离地球束缚，成为环绕太阳运动的“人造行星”；

4）当v≥v3时，被发射物体将从太阳系中逃逸。

* 1. 地球半径

地球的直径有12742公里。地球赤道半径6378.137千米，极半径6356.752千米，平均半径约6371千米。所以地球的平均直径是2×6371=12742千米。地球是一个赤道直径大，南北极直径小的不规则球体。

* 1. 卫星在轨运行

第一宇宙速度v1=7.9km/s，地球平均半径6371km，近地卫星轨道高度500km，卫星绕地一圈 2\*Π\*(6371+500)/7.9/60=91.0335分钟 （经常说的卫星绕地球一圈约为90分钟），卫星每秒转过的角度 360/(91\*0.335\*60)=0.0659° （所以卫星在轨运行，看似速度很快，实际因为地球很大，卫星每秒转过的角度就很小，这就是星敏为什么能正常工作的原因）。

* 1. 卫星的轨道

卫星的轨道分为低中高三种，低轨道为 200km-2000km 之间，中轨道为 2000km20000km 之间，超过 20000km 的为高轨道。

中高轨道所能提供的通信能力有限， 主要作为地面通信的补充和延伸，低轨道卫星则能提供全球性的移动互联网服务，目前申请的低轨卫星星座大多位于 700km-1500km

* 1. 自恢复保险丝
  2. 电源转换芯片

LDO

DC-DC

* 1. 姿态

俯仰、偏航、滚转

* 1. 四元数转三个欧拉角
  2. 标准差

标准差是方差的算术平方根，用δ表示，标准差是反映一组数据离散程度最常用的一种量化形式，是表示精确度的重要指标

1. 硬件
   1. 电阻封装

贴片电阻常见的封装 0402 0603 0805 1206 1210 1812 2010 2512

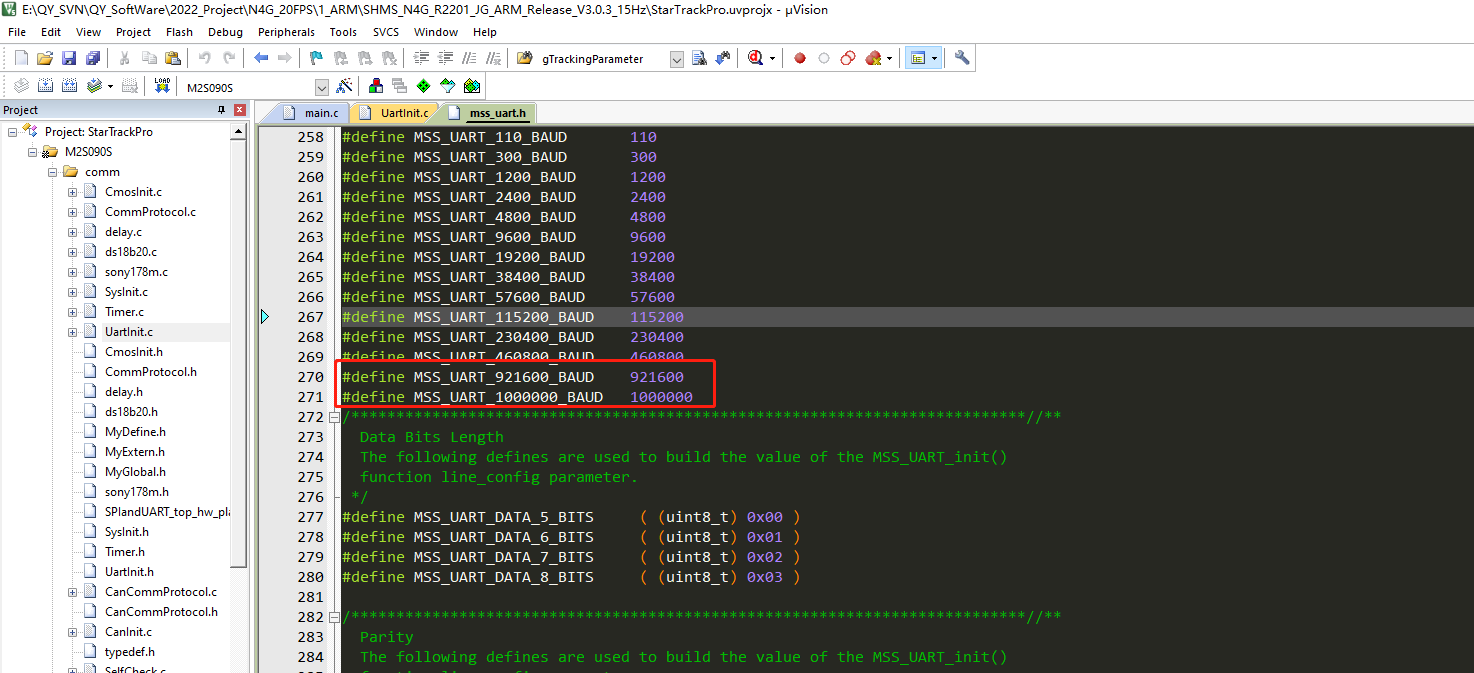
* 1. 绝缘和连通性

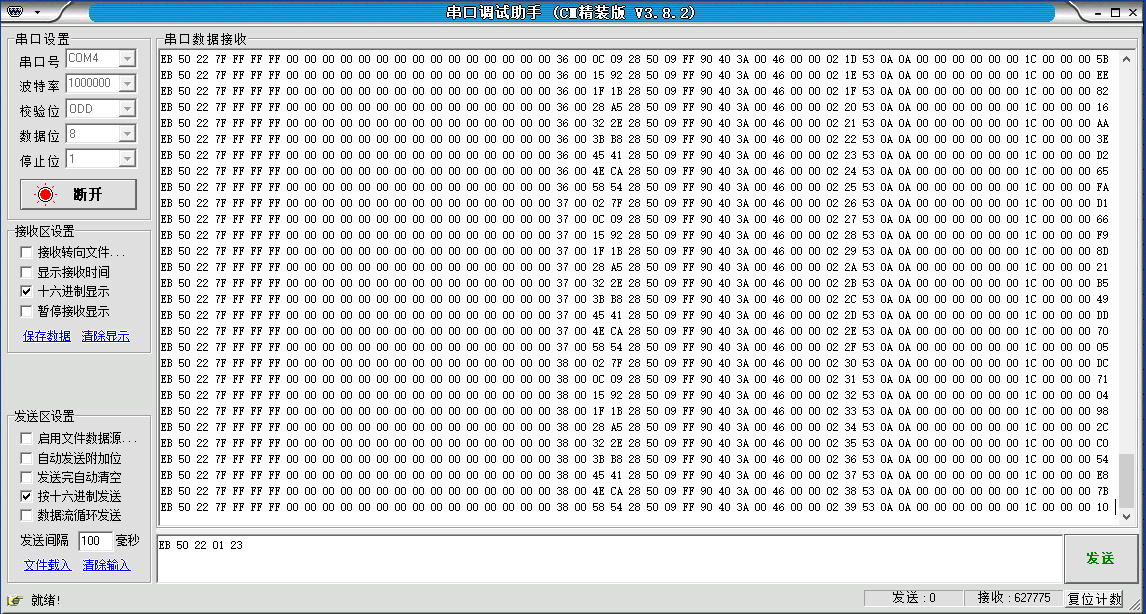
绝缘性测试阻抗 ∞

连通性测试 阻抗<0.3Ω

1. 软件调试
   1. 波特率测试

验证ARM软件初始化串口的波特率到921600bps 1Mbps





问题总结

只需要在.h文件中添加宏定义即可，刚开始发现波特率增大后，指令接收异常，换了一根串口线后，正常

加大波特率后，出现误码有可能和线有关系（注意）