

# RBH\_Matlab 编程说明

**2009-10-29 VER1.0**

Matlab 是美国 MathWorks 公司出品的商业数学软件，用于算法开发、数据可视化、数据分析以及数值计算的高级技术计算语言和交互式环境。它在数学类科技应用软件中在数值计算方面首屈一指。Matlab 可以进行矩阵运算、绘制函数和数据、实现算法、创建用户界面、连接其他编程语言的程序等，主要应用于工程计算、控制设计、信号处理与通讯、图像处理、信号检测、金融建模设计与分析等领域。

北京瑞博华公司的板卡全面支持 Matlab 的编程，而且编程方法非常简洁、为您快速开发应用系统创造条件。本公司提供完整的编程实例和详尽的说明，以及全汉字的软件注解，还有本公司提供全面的编程技术与硬件技术服务，这些都为您的开发铺平了道路。

由于采用了相同的接口方式，本例程对北京瑞博华公司的全部产品都适用。

为了便于您理解和应用，本说明主要以实例为基础来说明在 Matlab 下的编程方法，本实例的开发环境是 Matlab 2007b。

## 一、 编程实例的主要文件

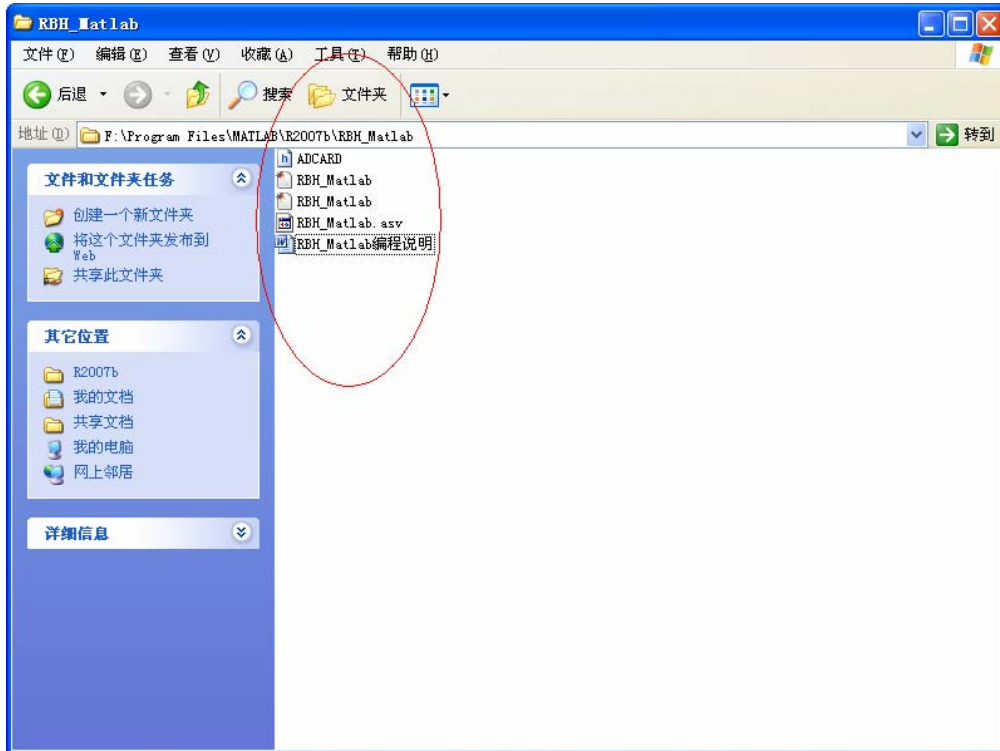


图 1 编程实例文件目录

如图 1 所示，目录中包含 5 个文件，这 5 个文件分为三类，分别介绍如下：

第一类：北京瑞博华公司提供的接口程序：

ADCARD.H：这是瑞博华公司专为 Matlab 提供的头文件，通过该文件，就可以实现对瑞博华公司提供的函数库进行操作，该文件应该包含在用户的应用程序中。

第二类：设计文件：

r8271.fig：用户界面文件，实现主要功能

r8271.m：应用软件源程序，实现主要功能

r8271.asv：程序运行时的文件

第三类：说明文件：

说明.txt：该文件记录软件的基本情况

## 二、 编程实例的开发过程

开发本软件的关键是在程序中调用瑞博华公司提供的 DLL 文件，然后就可

以如同正常的 Matlab 软件的开发。在 matlab 中利用动态链接库接口技术通常需要完成以下 4 个步骤：

1. 打开动态链接库文件。
2. 为调用函数准备数据。
3. 调用动态链接库文件中导出的函数。
4. 关闭动态链接库文件。

打开动态链接库函数：

```
Loadlibrary('mydll', 'hfile');
```

其中 mydll 是字符串变量，代表所要使用的 DLL 文件的名称。hfile 也是字符串变量，代表包含该 DLL 文件中所导出的函数定义的头文件名。hfile 包含以 C 语言定义的函数特征，对于本来就是以 C/C++开发的 DLL 文件，可以直接使用原来的头文件。对于其他语言开发的 DLL 文件，需要手工编写相应的 C 语言头文件。

调用动态链接库文件中导出的函数：

```
Calllib ('libname', 'funcname', arg1, arg2,.....argN)
```

其中 libname 是字符串变量，表示载入的 DLL 文件，它通常是载入的 DLL 文件名；funcname 也是字符串变量，代表需要调用的函数名；arg1, arg2....argN 则是调用函数 funcname 所需要的所有参数。

动态链接库文件的卸载：

```
Unloadlibrary ('libname');
```

其中 libname 为字符串变量，表示要卸载的 DLL 文件。

Return Type	Name	Arguments
[int32, uint16Ptr]	ADResult	(uint16Ptr)
[int32, structADResultPtr]	ADResultRecent	(structADResultPtr)
single	ChannelFrq	(int32, int32)
[cstring, int32Ptr, int32Ptr, int32Ptr, int32Ptr, singlePtr, singlePtr, int32Ptr]	ConfigInfo	(cstring, int32Ptr, int32Ptr, int32Ptr, int32Ptr, singlePtr, singlePtr, int32Ptr)
int32	GetADPropStrSize	
[int32, cstring]	GetADPropString	(cstring)
int32	GetLastError	
[int32, structADResultPtr, uint32Ptr]	GetSnapshot	(structADResultPtr, int32, uint32Ptr)
[int32, cstring, cstring]	IOCtl	(int32, cstring, int32, cstring)
int32	Initial	(uint16, uint16, uint32, uint16)
uint16	QueryBuf	
[int32, voidPtr, voidPtr]	RegisterNotify	(uint32, voidPtr, uint32, uint32, voidPtr)
uint32	StartIntr	(uint16, uint32, uint16, uint16, uint32, uint32, uint16)
uint32	StartSnapshot	(uint16, uint32, uint16, uint16, uint32, uint32, uint16, uint32)
int32	StopIntr	
uint16	inport	(uint16)
uint16	inportb	(uint16)
uint32	inportdw	(uint16)
	outport	(uint16, uint16)
	outportb	(uint16, uint16)
	outportdw	(uint16, uint32)

图 2 Adcard 动态链接库中所包含的函数名称、返回值类型以及参数类型

如图 2 所示，在初始化动态链接库后，可以通过 `libfunctionsview adcard` 命令来查看当前动态链接库中所包含的函数名称、返回值类型以及函数的参数类型，为下一步函数的调用提供方便。

### 三、实例程序的功能

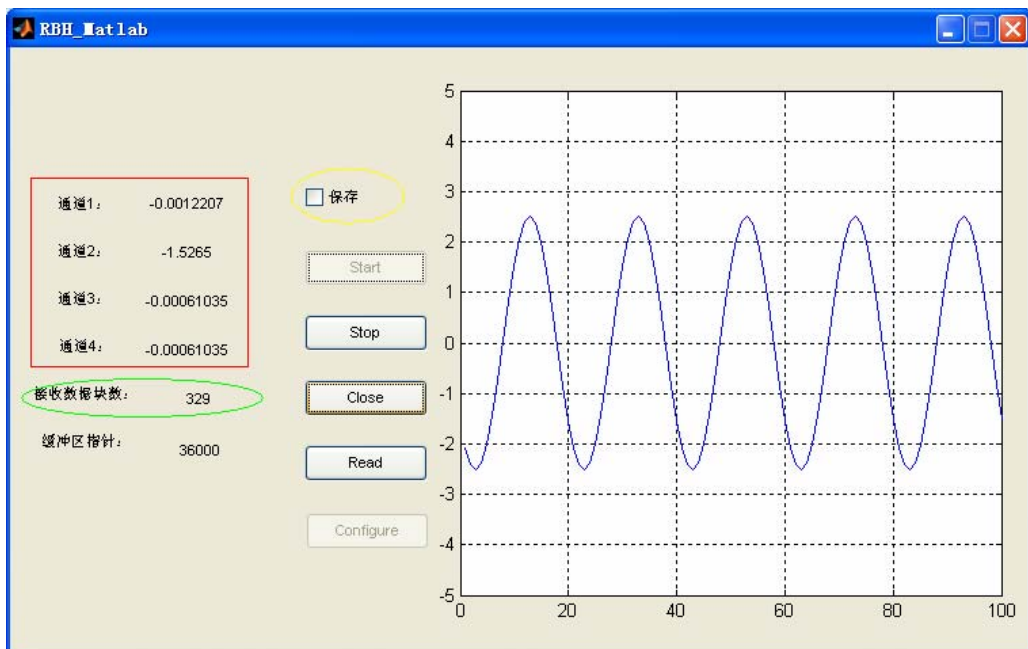


图 3 实例程序的功能

图 3 所示为本程序运行的界面，可以看出，例程实现了波形曲线显示与 AD 结果显示。

针对图 3，各个部分的功能的如下：

红色区域同时显示各个通道的电压值，本例程以 4 个通道为例。

绿色区域显示当前采集的数据块数。该数据不应该达到用户软件设定的内存缓冲区数，如果达到了用户设定的缓冲区数，表明可能产生数据丢失，解决的方法是增加内存缓冲区块数 NumBuf 这个变量的值，或增大 NumSamp 这个参数。

黄色区域表示保存数据功能，在用户采集过程中，当选中这个选项时，可以自动将采集到的数据保存在文件当中，文件名以日期时间默认命名，用户可以在保存后自行修改。

五个按钮分别代表开始采集、停止采集、关闭程序、读取保存的数据以及配置信息。该例程的一个功能就是可以保存、读取数据，方便用户记录。

图 3 波形的曲线显示，将采集到的 AD 值以波形的方式表现出来，直观而且容易理解。

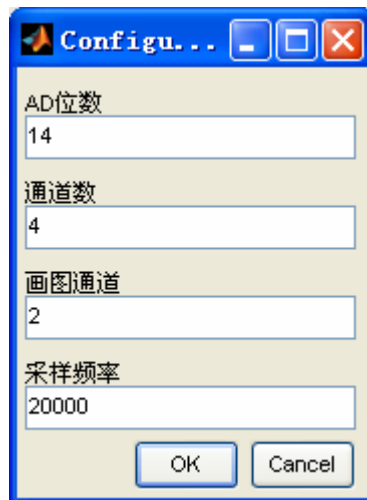


图 4 配置信息对话框

图 4 为配置信息对话框，用户可以在停止采集后打开此对话框对采集程序进行信息配置，主要包括所用采集卡的 AD 位数，所用采集卡的通道数，在主界面画图的通道号以及采样频率。

#### 四、 实例程序说明

实例代码总体结构非常简单，明了，并且有详细的注解。

##### 1. 数组与变量的定义

```
global NumBuf
global NumSamp
global begchn
global NumChn
global FrqSamp
global FrqFilter
global AmpGain
```

其中，NumBuf 为缓冲区个数，NumSamp 为每个缓冲区采样点数，一个采样点是指所有的通道采样一次，begchn 为起始通道，NumChn 为通道数，FrqSamp 为采样频率，FrqFilter 为滤波器频率，AmpGain 为放大器增益，本程序借用这个产生控制采集模式，当该参数=0 时，表示 AD 结果为未校准结果，否则为校准结果。正常使用时应该为 1。

```
handles.ADBuf=1:6000;
shortdata=uint16(handles.ADBuf);
handles.lp=libpointer('uint16Ptr',shortdata);
```

定义数据缓冲区，handles.lp 为指向缓冲区数组的指针。

```
global RecordPtr %数据指针
```

```
global TotalBuf %数据块数
```

## 2. 初始化采集参数

```
NumBuf = 10;
NumSamp = 1000;
begchn = 0;
NumChn = 5;
FrqSamp = 20000;
FrqFilter = 0;
AmpGain = 1;
```

### 3. 调用 adcard.dll

```
h=[' F:\Program Files\MATLAB\R2007b\work\adcard.h']; %头文件所在的地址  
loadlibrary('adcard',h); %调用adcard.dll，将dll文件包含到程序中。
```

### 4. 定义定时器

由于Matlab的gui编程组件中没有现成的定时器空间，所以在使用定时器之前要先定义定时器。

```
handles.timer=timer('Period',0.2,'ExecutionMode','FixedRate','TimerFcn',{ @Msg  
update,handles});
```

```
handles.timer1=timer('Period',0.2,'ExecutionMode','FixedRate','TimerFcn',{ @Dr  
aw,handles});
```

这两个定时器一个负责采集数据的读取，第二个负责读取已存数据时的画图。

### 5. 定时器程序

定时器程序是高速连续采集的关键程序。

```
ChnNum=1;%要画在面板上的通道号
```

```
shortdata=1:6000; %定义临时缓冲区
```

```
shortdata(1:6000)=0;
```

```
NumFill=calllib('adcard','QueryBuf'); %填满的缓冲区数
```

```
if NumFill==0
```

```
    return;
```

```
end
```

```
for i=1:NumFill
```

```
    j=calllib('adcard','ADResult',handles.lp); %获得采集数据
```

```
    shortdata=get(handles.lp,'value'); %将采集数据放入临时数组中
```

```
    if j==1
```

```
        RecordBuf(1:NumSamp*NumChn)=shortdata(1:NumSamp*NumChn); %
```

将采集数据放入大缓冲区数组

```

RecordPtr=RecordPtr+NumSamp*NumChn; %数据指针
if RecordPtr>=RecordBlock*NumSamp*NumChn
    RecordPtr=0;
end
%画图
for i=1:NumSamp
    DrawBuf(i)=RecordBuf(1+NumChn*(i-1)+ChnNum); %ChnNum是
要画的通道号
    DrawBuf(i)=(DrawBuf(i)-VZero)/VMax*SpainV;
    if fFlag==1
        count=fwrite(fid,DrawBuf(i),'double');
    end
end
% 将数据画在面板上
plot(DrawBuf(901:1000));
axis([0 100 -5 5]);
grid on
end
end
TotalBuf=TotalBuf+NumFill;
%显示采集的数据块数以及当前数据指针
set(handles.lblBlock,'string',num2str(TotalBuf));
set(handles.lblptr,'string',num2str(RecordPtr));
%显示4个通道的电压值
for i=2:NumChn
    ad_chn(i)=shortdata(1 + (NumSamp - 1) * NumChn + i);
    if ad_chn(i)<0
        ad_chn(i) = ad_chn(i)+65536;
    end
    v_chn(i) = (ad_chn(i)-VZero)/VMax*SpainV;

```

```

end

set(handles.text9,'string',num2str(v_chn(2)));
set(handles.text10,'string',num2str(v_chn(3)));
set(handles.text11,'string',num2str(v_chn(4)));
set(handles.text12,'string',num2str(v_chn(5)));

```

定时器程序

完成对已存数据的画图

NumSamp=1000;

```

if Rflag==0
    [DrawBuf count]=fread(fid1,NumSamp,'double');%从文件中读取数据
else
    return
end

if count<NumSamp
    fclose(fid1);
    Rflag=Rflag+1;
    return
end

DrawBuf=DrawBuf'; %获得的数据进行转置
%画图
plot(DrawBuf(901:1000));
axis([0 100 -5 5]);
grid on;

```

6. 开始采集的命令

```

calllib('adcard','StopIntr'); %先停止采集卡

i = calllib('adcard','Initial',0,0,0,0);%初始化采集卡

if i~=1

    return;

```

end

%开始采集

```
i=calllib('adcard','StartIntr',NumBuf,NumSamp,begchn,NumChn,FrqSamp,FrqFilter,  
AmpGain);
```

%重新定义缓冲区的大小

```
handles.ADBuf=1:(NumSamp*NumChn+1);
```

```
RecordBuf=1:(RecordBlock*NumSamp*NumChn);
```

```
shortdata=uint16(handles.ADBuf);
```

```
handles.lp=libpointer('uint16Ptr',shortdata);
```

%开启定时器

```
start(handles.timer);
```

%完成按钮的隐藏与显示功能

```
set(handles.StartButton,'enable','off');
```

```
set(handles.StopButton,'enable','on');
```

## 7. 停止采集的命令

```
set(handles.StopButton,'enable','off');
```

```
i=calllib('adcard','StopIntr'); %停止采集卡
```

```
if i~=1
```

```
    return;
```

```
end
```

```
stop(handles.timer); %关闭定时器
```

```
set(handles.StartButton,'enable','on');
```

```
set(handles.checkbox1,'value',0);
```

%如果保存功能开启，将通知用户文件保存的位置

```
if fFlag==1
```

```
    fclose(fid);
```

```
    msgbox(['The data has been saved as ' Fname])
```

```
end
```

## 8. 关闭程序的命令

```
global Flag;

stop(handles.timer); %关闭定时器

stop(handles.timer1);

i=calllib('adcard','StopIntr'); % 停止采集卡

if i~=1

end

unloadlibrary adcard; %卸载dll文件

if Flag==1

    close Figure 1; %关闭画图面板

end

%将程序中的全局变量清除

clear all;
```

## 9. 保存数据的功能

%获取当前的日期时间，并以此打开一个二进制文件，准备写入数据

```
Fname=datestr(now);

Fname=strrep(Fname,' ','-');

Fname=strrep(Fname,':','-');

fid=fopen(['d:\' Fname '.dat'],'a+');

Fname=['d:\' Fname '.dat'];

fFlag=1;
```

## 10. 读取数据的命令

```
stop(handles.timer1);

Rflag=0;

NumSamp=1000;

[filename dirname]=uigetfile('.dat','Select the dat-file'); %打开一个对话框，获取文件
```

的目录和名称

```
if isequal(filename,0)
    return
end
fid1=fopen([dirname filename], 'r'); %打开文件
Flag=1;
start(handles.timer1);%开启定时器，将数据画到面板上
```

## 五、 进行 Matlab 编程时应该注意的问题

通过实验证明，采用北京瑞博华公司的硬件和软件，可以很好地实现与 Matlab 无缝连接，实现高速、连续的采集。

在编程时，有以下几点需要注意：

1. 采用 Matlab 编程时，一定要先调用 adcard.dll 文件，结束程序时，要卸载 adcard.dll 文件。
2. 配置参数 NumBuf, NumSamp 时，要考虑采样频率 FrqSamp，当 FrqSamp 很大时，NumSamp 就应该增大，当 FrqSamp 很小时，NumSamp 就应该变小。NumBuf 的要求是每次定时器读取得到的缓冲区数要小于 NumBuf，并且还有一定的余量。

## 六、 技术支持与服务

尽管我们努力使编程例程和文档尽可能地详尽，但还是有可能不能满足您的需要，请您及时与我们联系，并请经常光顾我们的网站[www.rbh.com.cn](http://www.rbh.com.cn)，在该网站上会经常有新的产品或新的文档发布，这些也许对您的开发有帮助。