**EEG/EMG（ADS1299）采集系统上位机程序说明**

**更新于2020.4.18**

该程序使用QT C++编写。

目录

[第一章 无线连接和接收数据格式约定 3](#_Toc10562281)

[1.1组网相关 3](#_Toc10562282)

[1.2 数据格式： 3](#_Toc10562283)

[1.2.1 数据基础： 3](#_Toc10562284)

[1.2.2 数据包格式约定： 3](#_Toc10562285)

[1.2.3通道数据恢复 4](#_Toc10562286)

[1.3 PC对电路板的控制： 4](#_Toc10562287)

[第二章 数据存储以及存储格式说明 5](#_Toc10562288)

[2.1 存储路径 5](#_Toc10562289)

[2.2 存储格式 5](#_Toc10562290)

[第三章 GUI界面功能说明 6](#_Toc10562291)

[3.1 通道数选择以及使能 6](#_Toc10562292)

[3.2 绘图 6](#_Toc10562293)

[3.3 基础滤波功能 7](#_Toc10562294)

[3.3.1高通滤波器滤除直流 7](#_Toc10562295)

[3.3.2 陷波滤波器 7](#_Toc10562296)

[3.3.3 带通滤波器 7](#_Toc10562297)

[3.4 FFT计算 7](#_Toc10562298)

[3.4.1 fft计算 7](#_Toc10562299)

[3.4.2 存在问题 8](#_Toc10562300)

[3.5 Glazer评估 8](#_Toc10562301)

[3.5.1 使用流程 8](#_Toc10562302)

[3.5.2 存在问题 8](#_Toc10562303)

[3.6 板子调试与配置功能 9](#_Toc10562304)

[3.7 短时傅里叶变换功能 9](#_Toc10562305)

[3.7.1 流程 9](#_Toc10562306)

[3.7.2 存在问题 9](#_Toc10562307)

[3.8 MARK功能 9](#_Toc10562308)

[3.9 读取已存储文件绘图 9](#_Toc10562309)

# 第一章 连接和接收数据格式约定

## 1.1组网相关

（1）Wi-Fi

采用TCP协议

上位机作为服务器端，侦听端口。端口号默认为61613。可以修改IP以及端口号。

组网方式为：PC和电路板通过Wi-Fi连接上路由器（默认情况下必须使用TP-LINK系列路由，板子默认寻找地址为192.168.1.100，需保证PC分配到该地址）。路由SSID为OpenBCI，密码为12345678 （115200配置）

(2) 串口

可以更改串口号，停止位等，具体见GUI配置页面。

## 1.2 数据格式：

### 1.2.1 数据基础：

板子ADC为24bit，即单通道的采样数据为3字节。板子将发送4字节数据，格式为补码格式。

### 1.2.2 数据包格式约定：

数据包分为调试数据和采集数据。调试数据为硬件的启动Debug信息，用于调试。采集数据为ADS1299采集数据。

1. 调试数据

4字节包序号+4字节标识+（X字节）调试信息+ 1字节累加和

|  |  |  |
| --- | --- | --- |
| 字节数 | 内容 | 说明 |
| 4 | 包序号 | 从0开始计数 |
| 4 | 标识 | 第一字节和第二字节为包头判断的辅助标识，固定为0X5A,0XA5。第三字节为调试信号长度，0代表不是调试信号，非0代表后面的调试信息的字节数。最后一字节用于mark标识，1为按键触发，否则为0 |
| X字节 | 调试信息 | X字节调试信息，X由标识中的第三字节决定 |
| 1字节 | 累加和 | 前面所有数据的累加和 |

1. 采集数据

（1）32\*N通道：

4字节包序号+4字节标识+（4字节\*32通道+8字节脱落）\*N块32路板+ 1字节累加和

|  |  |  |  |
| --- | --- | --- | --- |
| 字节数 | 内容 | | 说明 |
| 4 | 包序号 | | 从0开始计数 |
| 4 | 标识 | | 第一字节和第二字节为包头判断的辅助标识，固定为0X5A,0XA5。第三字节为调试信号长度，0代表不是调试信号。最后一字节用于mark标识，1为按键触发，否则为0 |
| （4\*32+8）\*N | 4\*32字节 | 通道1~32采集数据 | 每通道4字节数据，补码形式，第一字节为最低位，第四字节为最高位。例如接收到数据0x76 ,0xBF ,0x12 ,0x00，则实际数据为0x0012BF76 |
| 8字节 | 通道1~32脱落检测数据 | 8个字节，64bit，分别为：ch1p, ch2p, ch3p ,..., ch32p, ch1n, ch2n ,..., ch32n  脱落为1，否则为0 |
| 4\*32字节 | 通道33~64采集数据 |  |
| 8字节 | 通道66~64脱落检测数据 |  |
| … | … |  |
| 4\*32字节 | 通道32\*(N-1)+1~32\*N采集数据 |  |
| 8字节 | 通道32\*(N-1)+1~32\*N脱落检测数据 |  |
| 1字节 | 累加和 | | 前面所有数据的累加和 |

（2）小于32通道时(假如为X通道)：

|  |  |  |  |
| --- | --- | --- | --- |
| 字节数 | 内容 | | 说明 |
| 4 | 包序号 | | 从0开始计数 |
| 4 | 标识 | | 最后一字节用于mark标识，1为按键触发，否则为0 |
| （4\*X+8） | 4\*X字节 | 通道1~32采集数据 | 每通道4字节数据，补码形式，第一字节为最低位，第四字节为最高位。例如接收到数据0x76 ,0xBF ,0x12 ,0x00，则实际数据为0x0012BF76 |
| 8字节 | 脱落检测数据 | 8个字节，64bit，分别为：ch1p, ch2p, ch3p ,..., ch32p, ch1n, ch2n ,..., ch32n  脱落为1，否则为0 |
| 1字节 | 累加和 | | 前面所有数据的累加和 |

### 1.2.3通道数据恢复

收到的4个字节数据恢复为单位为uV的幅度数据（24为默认放大器增益，所以不推荐修改放大器增益，否则该处数值需要对应修改）：

V = （收到的数据补码恢复为原码后）/24

该数据带直流分量，需要高通滤波滤除直流才会回到0V基准线。

## 1.3 PC对电路板的控制：

高bit先发。

下表列举部分命令，详细寄存器地址以及值参考ADS1299手册。

|  |  |
| --- | --- |
| 内容 | 命令 |
| 启动采样 | 0x aa 06 01 |
| 暂停采样 | 0x aa 06 00 |
| 读单次数据 | 0x aa 05 |
| 复位 | 0x aa 01 |
| 写寄存器 | 0x aa 03 addr value |
| 读寄存器 | 0x aa 04 addr |
| 改变通道数 | 0x aa 07 value |
| 切换传输模式 | 0x aa 08 01 串口  0x aa 08 02 wifi |
| 配置PWM波频率 | 0x aa 09 value |

# 第二章 数据存储以及存储格式说明

## 2.1 存储路径

数据存储默认在应用程序所在目录新建ZJUEEGDATA文件夹，可选择其他文件路径存储，如下图所示



## 2.2 存储格式

数据保存为文本文档，命名方式为“年-月-日[时-分-秒]-通道数-采样率”；数据格式为从板子接收到的数据原格式（补码）。即4字节包序号+4字节标识+（4字节\*32通道+8字节脱落）\*N块32路板+1字节累加和。详细说明见[1.2.2](#_1.2.2_数据包格式约定：)。

目前一次运行保存3个文件：

* 包含所有数据的二进制文件，“raw\_年-月-日[时-分-秒] -通道数-采样率”。
* 包含处理过后的通道数据，用空格分隔，“processed\_年-月-日[时-分-秒] -通道数-采样率”
* 包含调试数据的字符串文件。“debug\_年-月-日[时-分-秒] -通道数-采样率”

# 第三章 GUI界面功能说明

## 3.1 通道数选择以及使能

可在连接上板子后，修改通道数

选定后界面中TAB选项卡内显示各通道的绘图窗，每个TAB选项卡显示32通道数据。

程序默认使能前8个通道。可通过勾选/不勾选通道前面选择按钮来进行通道使能。注意，这里使能只是隐藏了绘图，并不是令板子特定通道停止采集。还可以通过菜单栏“配置”->“通道”来快速使能通道。

使能的通道可显示波形和对应实时FFT图线。

## 3.2 绘图

界面如下图所示（旧图，笔者懒得更新新图了），包含时域绘图区域，FFT绘图区域，以及坐标控制。



绘图点数与FFT点数将根据采样率变化。FFT固定为1Hz频谱分辨率以及1s刷新率。

时域绘图区Widget提升为自定义的myframe，重载了PaintEvent进行绘图。使用QT双缓冲机制，先将绘图内容写入pixmap中。

FFT绘图区Widget提升为qcustomplot，来自QT绘图库QCustomPlot，官网<https://www.qcustomplot.com/>可获得所有相关文档。

FFT坐标控制在菜单栏“配置”->“FFT配置”。默认为自动坐标系，可手动修改显示量程。

## 3.3 基础滤波功能

鉴于实际采集过程中出现过工频以及带外噪声幅度为mV级别的情况（尤其高接触阻抗场合），而双线性二阶滤波器无法在阻带形成60dB（mV抑制为uV）衰减，这里引用外部算法库（来源github，https://github.com/berndporr/iir1）进行滤波计算。

需在程序运行速度与滤波器截止频率处的衰减进行折衷，滤波器阶数不宜太高。

### 3.3.1高通滤波器滤除直流

8阶切比雪夫I型滤波器，直接II型结构缩减计算开支。截止频率1Hz，Ripple 1dB。

### 3.3.2 陷波滤波器

8阶切比雪夫I型滤波器，直接II型结构缩减计算开支。中心频率50Hz，带宽10Hz，Ripple 1dB。

### 3.3.3 带通滤波器

8阶切比雪夫I型滤波器，直接II型结构缩减计算开支。Ripple 1dB。

## 3.4 FFT计算

### 3.4.1 fft计算

FFT计算仅在32通道场合下可以使用，算法来源MATLAB的快速傅里叶函数fft(x)。实测64通道即可时波形显示卡顿，因此仅设置32通道时可用。通道高于32时也只计算前32通道的采样率。

## 3.5 Glazer评估

实现肌电Glazer评估功能。功能按键如下图所示：



### 3.5.1 使用流程

（1）使用时先勾选RMS功能。RMS将对时域数据添加滑动RMS窗口，窗长100，无重叠。勾选RMS功能后，开始、结束按键将使能。

（2）点击开始按键，空白处出现秒表计时，存储路径下将额外生成以“年-月-日[时-分-秒]glazer”命名的文本文档，用于存储按下开始按键之后，结束按键之前的数据。

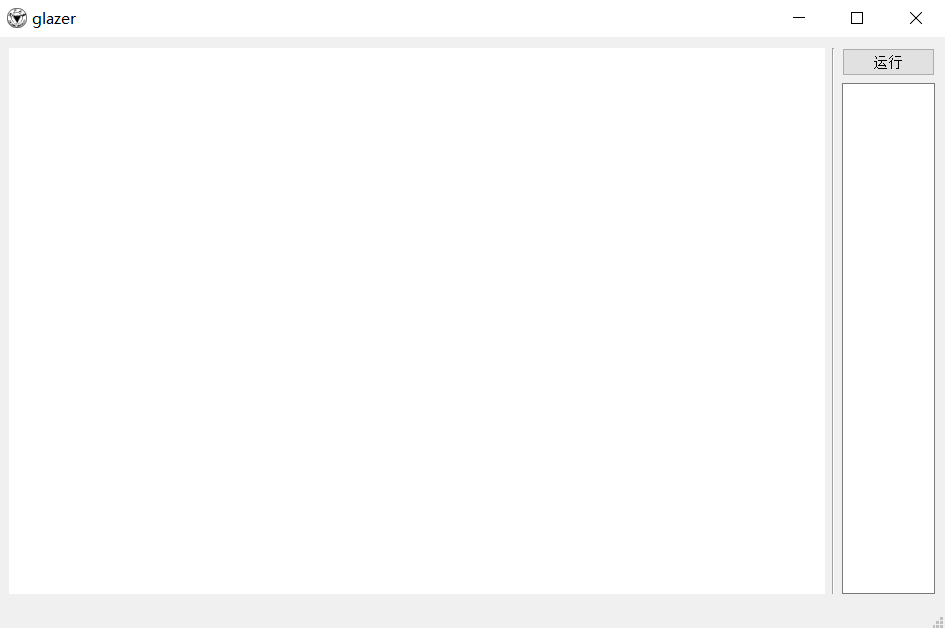
（3）完成glazer评估后，点击结束按键，停止glazer数据的存储。

（4）点击右侧箭头按钮，弹出glazer绘图界面，单击运行开始绘图。将会对数据进行（该功能运行速度较慢，推荐使用MATLAB分析）

32阶带通滤波（中心频率260Hz，通带带宽470Hz）；、

32阶陷波（中心频率50Hz，阻带带宽10Hz）；

滑动RMS处理（窗长200，重叠30）；



### 3.5.2 存在问题

（1）计算速度过慢，时间大部分消耗在数据解析上，即从补码中提取1通道double型数据。后期可考虑存储时即存储该double型数据。

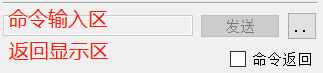
（2）计算速度过慢，第二耗时的是高达32阶的滤波器，降低至10阶也可以。

（3）滤波器截止频率、解码的通道数、RMS窗长和重叠数值应设置为可修改的，这里未实现。

（4）后续数据优化算法以及给出glazer评估数值均未实现。

## 3.6 板子调试与配置功能

该功能实现向板子下发命令，以及读取板子返回的数值。



（1）命令输入区可输入待发送的命令，格式为16进制，不区分大小写，不理会空格。例如AA 03 01；

（2）勾选命令返回选项后，可接收返回的数值。在返回显示区显示。格式为16进制字符。如A8；

## 3.7 短时傅里叶变换功能

该功能不可用！由于数据通常过大，申请的内存空间过大，导致无法使用。需要将数据进行截断处理。

### 3.7.1 流程

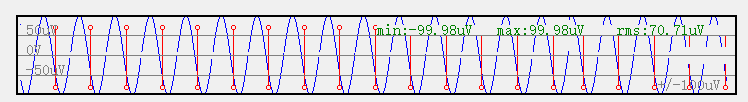
菜单栏“处理”->”STFT”启动STFT对话框，选择文件，配置好各参数后启动。

### 3.7.2 存在问题

使用外部库sp++。该库借鉴MATLAB算法，计算STFT时滑动窗重叠点数为（滑动窗长-1）。计算短时傅里叶变换时按数据量申请内存空间。由于数据量大，该行为会造成内存不足（远超1G）而导致申请内存失败，程序崩溃。目前实测1k采样率，1min以内数据可进行STFT计算。

## 3.8 MARK功能

接收到的数据第二字节标识的最后1bit为MARK标识，由板子按键触发，用于记录特殊时刻点。在图形上显示为红色竖杠。



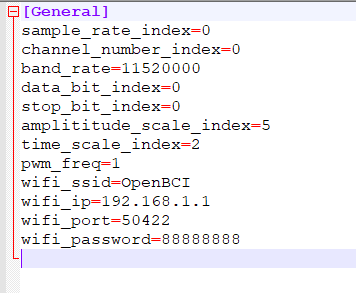
## 3.9 读取已存储文件绘图

菜单栏“文件”->“打开文件”选择对应文件后，点击运行按键即可开始绘图。绘图可选用各种滤波器，RMS。

经过多个版本更迭（需求总是在变），目前可以回放波形，但在读取结束时会卡死。这是由于没有判断结束的逻辑。且稳定性测试还未进行。

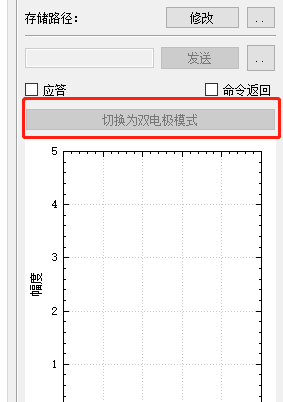
## 3.10 关于默认配置

由于需求总是在变！！！！笔者将部分默认配置写入配置文件，而不再是固定的。配置文件在工作目录下“config.ini”。第一次将使用内置的配置并生成config.ini。后续运行将读取配置。配置如下图：



## 3.11 关于单双电极切换

系统自带单双电极切换功能，且一键切换。



由于切换时需要发送的命令得不到确切答案。笔者将其配置命令单独放置在工作目录下的文本文档中，可以随时更改。



配置命令省略AA 03

## 3.12 关于应答机制

该设计仿照TCP中的三次握手，上位机发出命令，MCU对每条命令都需要回复。如果连续多次重发命令失败，则会弹窗报错。该功能在connect.cpp中，搜索关键词answer即可找到对应代码。目前应答机制重发5次，不同命令间隔50ms，重发间隔也是50ms。应答机制可以通过界面上的复选框来选择开启与否。



注：该功能笔者并不情愿添加，因为该功能是用于补救连接不稳定，数据丢失的问题。正常情况下，TCP连接以及串口连接不应该有如此高的数据丢失率，但确实这样发生了，笔者不得已采用了该措施。由于无法区分采样数据和应答数据，笔者只得相信传输延时如预期一样，按先后顺序处理数据，实际上后期是会发生问题的。一种解决的方法是重新约定数据帧结构，区分开采集数据和回复数据。但根本还是解决数据丢失问题。数据丢失原因可能为：（1）MCU读取缓存有漏洞，造成数据丢失。（2）wifi模组性能问题，该wifi模组的丢包率远高于之前笔者所使用的wifi模组。（3）网络链路不稳定。

## 3.13 PWM波

该功能用于控制硬件上生成的正弦波频率。功能简单，代码一看就懂。

## 3.14 debug窗口

Debug窗口可以实时显示调试信息。与绘图界面为多线程显示，一次可以同时操作。但要明白一点，所有绘图相关的操作都会回到主线程执行。

# 第四章 代码说明

该说明为简易的说明，代码中已经注释很多了。

4.1 mainwindow.cpp

主要功能。包括默认配置，窗口建立，槽函数响应。

4.2 myframe.cpp

波形绘图。为了实现自定义的绘图功能，没有采用外部库。因此如果想修改波形绘图的话，需要看这个文件。

4.3 signalprocess.cpp

数据处理。包括FFT，滤波，统计值计算，rms计算。因此如果想要修改数据处理的话，需要看这个文件。

4.4 connect.cpp

连接和数据接收。包括wifi连接，串口连接，数据接收，数据处理。因此，如果想要修改数据收发，需要看这个文件。

4.5 stft.cpp

Stft窗口建立以及处理逻辑

4.6 channelset.cpp

通道数变更窗口建立以及处理逻辑

4.7 debugwidget.cpp

调试窗口建立以及显示。

4.8 fftset

FFT配置窗口建立

4.9 glazer.cpp

Glazer窗口建立以及处理逻辑

4.10 Iir.h

外部滤波器库

4.11 Portset.cpp

串口设置窗口建立

4.12 Wifiset.cpp

Wifi设置窗口建立。

4.13 usbhandle.cpp

该文件无需使用了。