

ساختن رابط تصویری دستگاه ثبت سیگنال عضله

راحیل سروش مژدهی، سینا جوادزاده

۱۰ تیر ۱۳۹۵

۱ توضیحات کلی در مورد برنامه

توضیحات نحوه کارکرد برنامه به صورت کامل در فیلم داده شده است در نتیجه ما از تکرار آن خودداری می کنیم. نکات زیر را به عنوان مکملی بر فیلم آورده ایم. (فیلم در دراپ باکس آپلود شده و لینک آن در فولدر Video ذکر شده است)

1.1 Threads

برنامه به صورت کلی شامل دو thread است. Thread اصلی شامل یک تایمر نیز می باشد که در زمان سر ریز، تابع رسم دیتا (plot func) را صدا می زند. Thread فرعی نیز برای دریافت اطلاعات سریال و ذخیره کردن اطلاعات (در صورت تمایل کاربر) می باشد.

2.1 Modes

پس از برقراری ارتباط سریال و شروع دریافت داده، برنامه می تواند در ۲ مد کاری مختلف کار کند که توسط کاربر قابل انتخاب است.

1.2.1 Auto Detection

این مد صرفاً برای تشخیص انقباض عضله دست به سمت جلو یا عقب است، به این صورت که دیتاها به صورت زنده نمایش داده می شوند و به محض تشخیص یک حرکت عکس مربوطه برجسته شده و سبز می شود و در این حالت به مدت یک ثانیه می ماند. این تشخیص توسط یکی از تحلیل های عددی اشاره شده در بخش بعد انجام می شود. قابل توجه است که در این حالت استفاده از دکمه های Pause یا Resume امکان پذیر نیست و تشخیص حرکت به عهده ی خود برنامه است، اما Record کردن با استفاده از دکمه های مربوطه بلامانع است و حتی ۱ ثانیه ایست زمانی، خللی در این فرایند بوجود نمی آورد (به علت جدا بودن Thread ها).

Manual ۲.۲.۱

این مد برای مشاهده و تحلیل دقیق سیگنال EMG توسط خود کاربر به وسیله ابزارهای متنوع و قدرتمندی است که در برنامه در اختیارش گذاشته شده است. در این مد دکمه های Pause و Resume قابل استفاده بوده و رسم سیگنال به صورت زنده فقط و فقط توسط کاربر متوقف می شود. پس از متوقف کردن نمودار توسط کاربر بوسیله دکمه ی Pause پنجره ای بر روی نمودار پدیدار می شود که برای کاربر این امکان را فراهم می آورد که محدوده ی دلخواه خود را از سیگنال جدا کرده و به تحلیل آن پردازد (توضیحات کامل در فیلم داده شده). حال با انتخاب تب Analyse Data کاربر میتواند پنجره ای از سیگنال که انتخاب کرده را تحلیل کند. تحلیل ها به دو دسته ی عددی و نموداری تقسیم شده اند. تحلیل های عددی در برنامه با استفاده از Cython پیاده سازی شده زیرا عملیات باید روی ۲۰۰۰ عدد دیتا انجام شود و استفاده از Cython سرعت را به مراتب بالا می برد. اما تحلیل های نموداری به علت نداشتن حلقه و استفاده مکرر از توابع کتابخانه SciPy قابل بهینه سازی نبوده و در پایتون پیاده سازی شده اند. فرمول ها و اسم کامل تحلیل ها در برنامه آمده است. فرمول ها برای قسمت تحلیل عددی به علت پشتیبانی نشدن latex توسط ویجت label به صورت عکس پیاده سازی شده اند. قابل ذکر است که Record کردن در این مد هم بلامانع بوده و با Pause متوقف نمی شود.

PyQtGraph ۳.۱

کلیه نمودارها در Plot Widget پیاده سازی شده و توسط کتابخانه ی PyQtGraph رسم می شوند. این کتابخانه برای سرعت های بالای رسم سیگنال به ویژه به صورت Live استفاده می شود و دارای امکانات فراوانی است. زوم کردن و جابه جایی پنجره به راحتی قابل انجام است. این نمودارها قابلیت ذخیره در فرمت های گوناگون را نیز دارند.

Enable Set ۴.۱

در پیاده سازی دکمه ها از روش Enable و Disable کردن استفاده شده تا از گذاشتن شرط های اضافی در برنامه جلوگیری شود.

Qpixmap ۵.۱

عکس ها در ویجت label و با دستور Qpixmap ساماندهی می شوند.

۶.۱ Mouse Move

در نمودار رسم شده در تب تحلیل داده یک خط عمومی اضافه شده، که با حرکت ماوس حرکت می کند و مقدار دقیق سیگنال ها را در آن موقعیت خاص نشان می دهد.

۲ روند پیشروی پروژه تا رسیدن به شکل نهایی آن

بزرگترین مشکلی که در مسیر پروژه به آن برخوردیم مشکل رسم ۲ سیگنال به صورت موازی و با سرعت بالا و عدم ایجاد اختلال در دریافت دیتای سریال بود. زیرا سرعت دریافت داده زیاد بوده و باید بدون اختلال انجام می شد (۱۰۰ نمونه در ثانیه و هر نمونه ۵ بایت)

۱.۲ مرحله ی اول

در ابتدا ما این عملیات را با استفاده از کتابخانه matplotlib انجام دادیم. به صورت دو Thread مجزا (به غیر از Thread اصلی برنامه) یکی برای رسم و دیگری برای دریافت دیتا. اما با بالا بردن سرعت رسم برای نرم تر دیده شدن نمودار در حرکت علاوه بر بالا رفتن بیش از حد درصد پردازش، دریافت دیتا نیز با مشکل روبرو میشد و دیتا ها در صف می رفتند و تاخیر در دریافت دیتا بوجود می آمد.

۲.۲ مرحله ی دوم

در مرحله ی بعد به این نتیجه رسیدیم که برای حذف تاثیر مخرب رسم دیتا بر دریافت آن، پروسس دریافت دیتا را جدا کرده و برنامه را بصورت multiprocessing پیاده سازی کنیم. پس از انجام این روش نیز شاهد بهبود قابل توجهی در عملکرد برنامه نبودیم. زیرا در مرحله ی اشتراک متغیرها بین پروسس ها ما مجبور بودیم متغیرهایی با حجم بالا را به اشتراک بگذاریم که خود این کار باعث پایین آمدن کارکرد برنامه می شد. بنابراین این روش را کنار گذاشتیم.

۳.۲ مرحله ی سوم

در مرحله ی بعدی تصمیم گرفتیم از کتابخانه PyQtGraph به جای کتابخانه matplotlib استفاده کنیم که در اندازه گیری های اولیه حدود ۱۵۰ برابر سریع تر نمودار را بروزرسانی می کرد. اما با قرار دادن تابع رسم نمودار بوسیله ی PyQtGraph در یک Thread جدید به مشکل جدیدی مواجه شدیم. با کار هایی مثل کوچک و بزرگ کردن صفحه اصلی نمودار هنگ می کرد و دیگر حتی پاک هم نمی شد.

۴.۲ مرحله ی آخر

در نهایت به این نتیجه رسیدیم که باید رسم دیتا را بوسیله ی PyQtGraph اما این بار در Thread اصلی برنامه و با استفاده از یک Timer انجام دهیم تا هنگ نکند. این روش دارای سرعت بسیار بالا است و هنگ هم نمی کند و در دریافت دیتا های سریال نیز خللی ایجاد نمی کند. این روش رسم حتی این امکان را برای ما فراهم کرد تا سرعت ارسال داده را تا ۱۰۰۰ پکیج در ثانیه افزایش دهیم. که با این نرخ ما توانستیم تمام محتوای فرکانسی سیگنال EMG را پوشش دهیم. (محتوای فرکانسی سیگنال EMG از چند هرتز تا ۵۰۰ هرتز است)

۳ اجرای برنامه در لینوکس

برای اجرای صحیح برنامه در محیط لینوکس نیاز بود مراحل زیر را طی کنیم:

۱. نصب برنامه Anaconda در لینوکس

۲. نصب کتابخانه های مورد استفاده توسط Anaconda برای پایتون، مانند: کتابخانه های Qt و PyQt و ورژن ۵، کتابخانه ی PySerial برای ارتباط سریال، کتابخانه ی PyQtGraph و ورژن ۹.۱۱ که با Qt و ورژن ۵ سازگار است.

۳. درست کردن یک فایل جدید با پسوند .so. برای کد Cython زیرا کد قبلی با پسوند .pyd. در محیط لینوکس قابل import کردن در برنامه نبود. با یک برنامه setup.py به راحتی فایل .so. از روی فایل .pyx. ساخته شد. فایل setup استفاده شده در پوشه Cython stuff موجود است.

۴. وارد کردن پورت USB به داخل محیط VMWARE

۵. صادر کردن اجازه ی خواندن و نوشتن برای پورت مورد نظر بادرستور زیر: (ttyACM2 نام پورت USB ما است)

```
sudo chmod 777 /dev/ttyACM2
```

و در نهایت پس از طی کردن مراحل فوق برنامه به درستی قابل اجرا است. قابل ذکر است که با وجود اینکه محیط لینوکس ما داخل VMWARE بالا می آمد، هیچ گونه تاخیر یا اختلالی در اجرای برنامه و انتقال اطلاعات به وجود نیامد.